research center pharmaceutical engineering

# Diploma Thesis

(Diplomarbeit zur Erlangung des akademischen Grades eines Diplom-Ingenieurs der Studienrichtung Verfahrenstechnik an der Technischen Universität Graz)

# Modeling of the Hardening Process of Micro-Particles

Modellierung des Aushärtungsprozesses von Mikropartikeln

Hannes Pucher

Advised by:

Dipl.-Ing. Dr.-Ing. Daniele Suzzi

Dipl.-Ing. Dr.techn. Stefan Radl

Univ.-Prof. Dipl.-Ing. Dr.techn. Johannes G. Khinast

TU Graz
Graz University of Technology

Institute for Process and Particle Engineering

Graz, March 2011

Deutsche Fassung:
Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008
Genehmigung des Senates am 1.12.2008

## EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Graz, am …… ……..
(Unterschrift)

Englische Fassung:

## STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

…… ……..
date (signature)

**Acknowledgement**

I would like to take the opportunity to thank everybody who spent their time and shared their knowledge for helping me to complete my thesis. I am grateful for all the interesting and helpful discussions.

Foremost, I would like to express my sincere thankfulness to my supervisor Univ.-Prof. Dipl.-Ing. Dr.techn. Johannes G. Khinast who gave me the opportunity to write my diploma thesis on the Institute of Process and Particle Engineering and supported me during my work.

Besides my supervisor, I want to thank Dipl.-Ing. Dr.-Ing. Daniele Suzzi for his ongoing help and guidance through the development of my thesis. Especially I want to thank him for his supervision of my work and his feedback.
In addition, I am grateful for the many inputs from Dipl.-Ing. Dr. techn. Stefan Radl.
Furthermore, I want to thank Univ.-Prof. Dr.-Ing. habil. Günter Brenn and M. Sc. Nikolett Kiss for the excellent cooperation and all their inspiring contributions.

In addition I want to thank my colleagues who have helped in numerous ways especially Dipl.-Ing. Dr.techn. Heidrun Gruber-Wölfler, Dipl.-Ing. Michael Gruber, Johann Grubbauer, Michael Gruber and Johannes Hofer.

I would like to extend my thanks to the Research Center Pharmaceutical Engineering (RCPE GmbH.) for the financial support.

Finally I would like to express my deepest gratitude to my family. Without their encouragement and understanding it would have been impossible for me to finish my academic studies. Of course I want to thank all my friends for their support.

To all of you, thanks for always being there for me.

## Abstract

Considering the increasing complexity of drugs, process understanding in the pharmaceutical industry becomes more and more important. Often the method by which a drug is delivered has a significant influence on the therapeutic effectiveness. Biodegradable-controlled-release-systems, as investigated in this work, introduce new concepts in drug administration, can replace other dosage forms and provide new treatment options for doctors and patients in their fight against diseases.

Due the poorly reported process-engineering approaches the objective of this work was to investigate the extraction step of a micro-particle process on the basis of emulsification solvent extraction method. The gained insights should on the one hand provide a better understanding and on the other hand help to predict different characteristics of micro-particles.

Therefore a numerical model was developed and different experimental investigations were performed. The developed numerical model is capable to describe the mass transfer from the disperse phase into the continuous phase respectively the hardening/extraction process. At the first step the obtained numerical results were compared with an analytical solution, found by John Crank [7] which showed a good agreement. Furthermore a custom build shrinking model respectively a method to numerically compute the diameter reduction of the micro-particles was implemented.

Next the discretization uncertainty was investigated and in addition a parameter study was performed to find realistic ranges for the values of the unknown variables.

For the validation of the numerical model, the estimated results were compared with the experimentally determined diameter of the micro-particles. It showed, that the predictions of our numerical model fit experimental data reasonably well.

For the experimentally research of the process an experimental set up was build in the labs of the Research Center Pharmaceutical Engineering (RCPE GmbH) in Graz. The performed investigations were able to quantify the micro-particle size during the extraction process and determine the influences of different parameters on the final micro-particle quality.

## Kurzfassung

In Anbetracht der zunehmenden Komplexität von Medikamenten, gewinnt das Prozessverständnis in der pharmazeutischen Industrie immer mehr an Bedeutung. Oft hat die Methode, mit der ein Medikament zugeführt wird, einen entscheidenden Einfluss auf die therapeutische Effektivität. Biologisch abbaubare-kontrollierte-Freisetzungsysteme, die in dieser Arbeit untersucht wurden, könnten neue Konzepte in der Medikamentenverabreichung ermöglichen, andere Darreichungsformen ersetzen und neue Behandlungsmöglichkeiten für Ärzte und Patienten in ihrem Kampf gegen Krankheiten anbieten.

Aufgrund der kaum verfügbaren verfahrenstechnischen Konzepte, war das Ziel dieser Arbeit den Extraktionprozess eines Mikro-Partikel-Prozesses auf der Grundlage der „Solvent Extraction Method“ zu untersuchen. Die gewonnenen Erkenntnisse sollten einerseits das Prozessverständnis fördern und andererseits dazu dienen verschiedene Eigenschaften der Mikropartikel vorherzusagen.

Aus diesem Grund wurde ein numerisches Modell entwickelt und verschiedene experimentelle Untersuchungen durchgeführt. Das entwickelte numerische Modell ist in der Lage, den Stoffübergang von der dispersen Phase in die kontinuierliche Phase beziehungsweise den Aushärtungs- bzw. Extraktionsvorgang zu beschreiben. Im ersten Schritt wurden die erhaltenen numerischen Ergebnisse mit einer analytischen Lösung von John Crank [7] verglichen, welche eine gute Übereinstimmung zeigten. Darüber hinaus wurde ein selbst entwickeltes Schrumpfungs-Modell bzw. eine Methode zur numerischen Berechnung der Durchmesserreduktion der Mikro-Partikel implementiert.

Des Weiteren wurde der auftretende Diskretisierungsfehler untersucht und eine Parameter-Studie durchgeführt, um realistische Wertebereiche für die unbekannten Variablen zu finden. Zur Validierung des numerischen Modells wurden die errechneten Ergebnisse mit den experimentell bestimmten Durchmessern der Mikropartikel verglichen. Dies, zeigte dass Vorhersagen unseres numerischen Modells gut mit experimentell erhobenen Daten übereinstimmen.

Für die experimentelle Untersuchung des Prozesses wurde ein experimenteller Aufbau in den Labors der Research Center Pharmaceutical Engineering (RCPE GmbH) in Graz errichtet. Mithilfe der durchgeführten Untersuchungen konnte die Mikro-Partikelgröße während der Extraktion quantifiziert und die Einflüsse verschiedener Parameter auf die endgültige Mikro-Partikel Qualität ermittelt werden.

## Table of Contents

## List of Figures

## List of Tables

## Abbreviations

| | |
|---|---|
| SMD | Sauter mean diameter |
| CFL-number | Courant-Friedrichs-Lewy-number |
| EA | Ethyl acetate |
| BA | Benzyl alcohol |
| PVA | Polyvinyl alcohol |
| W | Water |

## Nomenclature

**Latin symbols:**

| | |
|---|---|
| $A_P$ | Surface area of the micro-particle [$m^2$] |
| $Bi$ | Biot number |
| $\Delta c$ | Concentration difference |
| $C_i$ | Initial concentration [kmol/$m^3$] |
| $C_0$ | Equilibrium concentration [kmol/$m^3$] |
| $C_S$ | Actual concentration [kmol/$m^3$] |
| $D_i$ | Diffusion coefficient of the component i [$m^2$/s] |
| $D_{i,CP}$ | Diffusion coefficient in the continuous phase [$m^2$/s] |
| $D_{i,DP}$ | Diffusion coefficient in the disperse phase [$m^2$/s] |
| $d_P$ | Diameter of the micro-particle [m] |
| $d_{P,t}$ | Characteristic length/dimension/diameter at time t [m] |
| $d_R$ | Diameter of the stirrer blade [m] |
| $Fo$ | Fourier number |
| $J_i$ | Volume flux relative to polymer velocity [$m^3$/($m^2$ s)] |
| $K$ | Equilibrium Constant |
| $M$ | Molecular weight [kg/kmol] |
| $\dot{M}_{t=o}$ | Mass transfer rate at the beginning [kg/s] |
| $m_{i,m,t}$ | Mean Mass of component i [kg] |
| $m_{Shell,m,t}$ | Mass of a spherical shell at time t [kg] |
| $\Delta m_{Shell,m,t}$ | Mass difference [kg] |
| $N_i$ | Number of drops |

| | |
|---|---|
| $n$ | Revolution rate of the stirrer [rpm] |
| $\dot{n}_i$ | Molar flow [mol/s] |
| $n_{Radial}$ | Number of radial sections |
| $R$ | Gas constant [J/(mol K)] |
| $Re$ | Reynolds number |
| $r$ | Radius of the micro-particle [m] |
| $r_{max}$ | Radius of the micro-particle [m] |
| $r_n$ | Normalized radius of the micro-particle |
| $r_{1,i}$, $r_{2,i}$, $r_{3,i}$ | System dependent constants for the component i |
| $Sc$ | Schmidt number |
| $Sh$ | Sherwood number |
| $T$ | Temperature [K] |
| $t$ | Time [s] |
| $t_{total}$ | Total extraction time [s] |
| $\Delta t$ | Time step [s] |
| $V$ | Molar volume of the dissolved substance at its boiling point [$cm^3$/mol] |
| $V_i$ | Volume of component i [$m^3$] |
| $V_{i,m,t}$ | Volume of a spherical shell [$m^3$] |
| $V_{Shell,r,t}$ | Volume of spherical shell at r and time t [$m^3$] |
| $V_{Sphere,m,t}$ | Volume of sphere at the time t [$m^3$] |
| $V_{Total}$ | Total volume [$m^3$] |
| $\Delta V$ | Volume difference [$m^3$] |
| $v$ | Mean fluid velocity [m/s] |
| $\bar{v}_i$ | Molar volumes of the component i [$m^3$/mol] |

| | |
|---|---|
| $v_3$ | Velocity of the polymer [m/s] |
| $w_i$ | Mass fraction of the component i |
| $w_{i,m,t}$ | Mean mass fraction of the component i |
| x | Association parameter |
| $\Delta x$ | Length of a radial section [µm] |

**Greek symbols:**

| | |
|---|---|
| $\alpha, \beta, \gamma$ | Ratios of the molar volumes |
| $\beta_i$ | Mass transport coefficient of the component i [m/s] |
| $\beta_{i,CP}$ | Mass transport coefficient in the continuous phase [m/s] |
| $\beta_{i,DP}$ | Mass transport coefficient in the disperse phase [m/s] |
| $\delta_i$ | Interface thickness [m] |
| $\eta$ | Dynamic viscosity of the solvent [1 cp = $10^{-3}$ Pas] |
| $\mu_i$ | Chemical potential of the component i [J/mol] |
| $\gamma_{Cp}$ | Kinematic viscosity of the continuous phase [$m^2/s$] |
| $\rho_{DP}$ | Density of the disperse phase [$kg/m^3$] |
| $\rho_i$ | Density of the component i [$kg/m^3$] |
| $\rho_{Shell,m,t}$ | Density of the spherical shell [$kg/m^3$] |
| $\varphi_{Analytical,i,r,t}$ | Analytically found volume fraction of the component i |
| $\varphi_i$ | Volume fraction of the component i |
| $\varphi_i^0$ | Initial composition of component i |
| $\varphi_{i,CP,\infty}$ | Volume fraction in the continuous phase |
| $\varphi_{i,DP,\infty}$ | Volume fraction in the disperse phase |
| $\varphi^*_{i,CP,I}$ | Equilibrium volume fraction at the interface of the CP |
| $\varphi^*_{i,DP,I}$ | Equilibrium volume fraction at the interface of the CP |
| $\varphi_{i,m,t}$ | Mean volume fraction of component i |
| $\varphi_{Numerical,i,r,t}$ | Numerically found volume fraction of the component i |
| $\varphi_{i,n,t}, \varphi_{i,n+1,t}$ | Volume fraction of component i at two specific radial positions |
| $\varphi_P$ | Volume fraction of the polymer |

| | |
|---|---|
| $\Delta\varphi$ | Volume fraction difference |
| $\chi_{ij}$ | Binary interaction parameters |

# 1. Introduction

## 1.1. Motivation

Micro-particles are widely used in the pharmaceutical industry as delivery forms for many kinds of drugs [16]. Microencapsulated drugs and vaccines have often a therapeutic benefit and that's why they forth the need to prepare such particles in larger quantities and in sufficient quality suitable for clinical trials and commercialization [9]. Generally, these so-called controlled-release systems refer to materials, or devices, which control the release time of a chemical, the release rate, or both [27].

In Figure 1-1 the concentration of the drug over the time is shown for a conventional and a controlled drug delivery system. In the conventional profile, at the beginning of the dissolution process the concentration of the drug increases very fast to a peak above the optimum therapeutic range. This peak could even lead to potentially fatal toxicity, for example in case of sleeping pills. After that, the concentration decreases rapidly under the optimum therapeutic range, leading no therapeutic effect.
As a consequence of these facts, it is evident that by using a conventional drug delivery system the time spent in the optimum concentration range is very short. The ideal concentration profile should be therefore nearly time independent and the optimal amount of drug should be release continuously. This "controlled" profile, also known as sustained release, is shown as well in Figure 1-1.

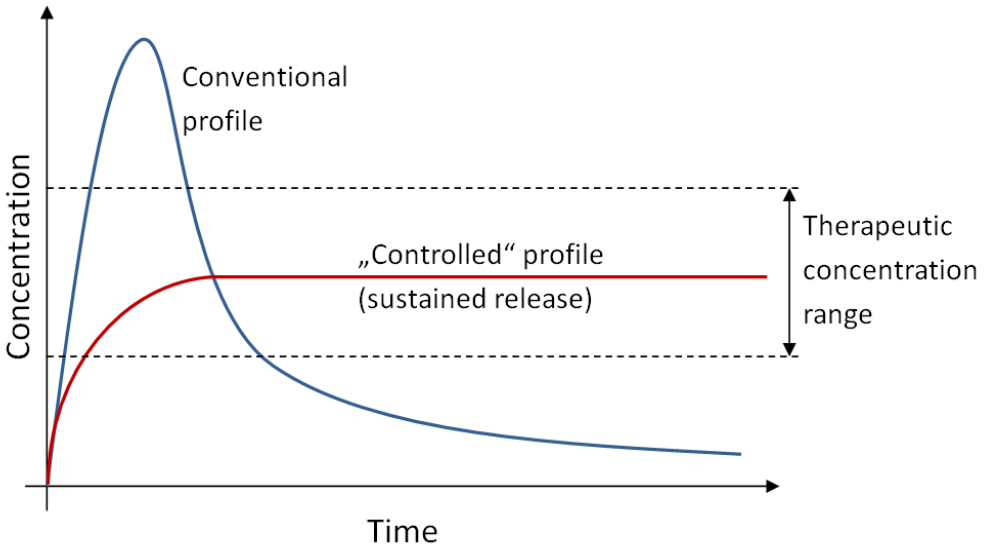



**Figure 1-1: Exemplary concentration c vs time t profile for conventional and controlled-release drug delivery devices [27]**

The influence of the method by which a drug is delivered on the therapeutic effectiveness becomes more and more important [3]. Without a continuous drug supply, the optimum therapeutic range of concentration of some drugs is attained only during very short time periods and concentration variations can be toxic or have no therapeutic benefit. Also in cases where a treatment over a longer time period is necessary, controlled-release systems are highly desirable. These systems could as well replace common dosages forms like multiple injections in order to ensure a certain drug level for a defined period of time [16].

Practically, the controlled profile could for example be achieved by incorporating the drug in a biodegradable polymeric carrier that controls the drug delivery and the release rate of the drug. These biodegradable-controlled-release-systems introduced new concepts in drug administration and usually provide high stability and easy handling. The desired release profile can be controlled by the choice and the production method of the polymer carrier. Common problems like short in vivo half-lives, physical and chemical instability, as well as low oral bioavailability of proteins, can be overcome by adopting sustained release concepts.

Historically, since the first appearance in the 1960s and 1970s, the number and variety of the controlled-released systems for drug delivery have increased dramatically. Nowadays these approaches have many different fields of application and are common in food, pesticides and cosmetic industry. Typical implementations are for example the release of flavors and vitamins in food, fragrances in perfumes, as well as inks in carbonless copy paper [27]. The largest application area for controlled release products is nevertheless the field of drug delivery in the pharmaceutical industry.

Despite the versatile use and the many advantages of the biodegradable-controlled-release systems, process-engineering aspects and numerical models of the different preparation techniques remain poorly reported [19]. A concrete numerical model concerning the solidification of microspheres is to our knowledge only build by Li et al. [18]. Actually, the numerical results of this model are only experimentally validated concerning the residual of solvent in the micro-particle. The resulting micro-particle size or other parameters were not compared.

To sum up, the ultimately idea behind a biodegradable-controlled-release system is to achieve more effective therapies with optimal dosage of the drug and provide new treatment options for the doctors and patients in their fight against diseases [24] [27].

Therefore, a deep understanding of each step of the production process is of great significance.

## 1.2. Goals

The final goal of the project, which this thesis is part of, is to develop a robust and scalable micro-particle process on the basis of solvent extraction method. This method is a very popular and often used technique for the preparation of micro-particles and is described in detail in Chapter 2. The final process should produce micro-particles with a defined (and close) size distribution and every micro-particle should consist of the same components.

The objective of this study is to systematically analyze the extraction step and to develop a numerical model for the description of the process. The final outcome is to predict different product characteristics, like the size of the final micro-particle, in dependence to the process parameters and the initial conditions of the extraction stage. The results and the knowledge gained in this work should provide on the one hand a better understanding of preparation process and, on the other hand, determine the influences of the most important process parameters on the final micro-particles. For this purpose, experimental analysis as well as computational mathematically modeling had been used.

For the detailed experimentally investigation of the process an experimental set up was built up in the labs of the Research Center Pharmaceutical Engineering (RCPE GmbH) in Graz. The mathematically model of the diffusion/extraction process was based on the MATLAB software from MathWorks Inc. Here, the most important phenomena simultaneously occurring inside the microparticle have been described in the numerical code.

Finally, the results obtained from the experimentally study were used to validate the mathematical model and a parameter study for different process parameters was carried out as well.

## 1.3. Thesis Outline

First, the state of the art for micro-particle preparation/microencapsulation processes used in the pharmaceutical industry are presented and described in Chapter 2. This background chapter should provide a general overview of the different preparation techniques and the common used chemicals.

The most important units of the experimental set up are then described in the Basic Engineering Chapter (Chapter 3).

The different conducted experiments and the different used measurement systems (HELOS and QICPIC) are accurately described in Chapter 4. Finally, the results of the performed experiments are also summarized in here.

Chapter 5 deals with the mathematical model for the description of the extraction/diffusion process. In this chapter the mass transport model, the shrinking model, the analytic verification, the used PDEPE solver and all the important parameters are described in detail.

In the following Discussion and Interpretation Chapter (Chapter 6) different parameters studies are performed. The objective is to get on the on hand a deeper understanding of the model and, on the other hand, to identify the ranges of some unknown parameters.

# 2. Background

## 2.1. Fabrication Techniques of biodegradable Micro-Particles for Drug Delivery Applications

According to Birnbaum et al. [5] there are innumerable methods to produce micro-particles, but the serviceability of a particular method is often determined by the solubility of the polymer and the drug. Furthermore, shear forces and temperature have often a damaging impact on the micro-particles when for example peptides and proteins have to be encapsulated [13].

Scheler [9] and Freitas et al. [28] defined that the production processes of polymer micro-particle are often modifications of three basic techniques, namely:

1. solvent extraction/evaporation,
2. phase separation (coacervation),
3. and spray drying.

According to Luan [20], the general requirements for micro-particle preparation are:

- Keep the stability of the encapsulated active ingredient.
- Achieve optimal drug loading, high encapsulation efficiency and yield.
- Obtain the desired drug release profile.
- Produce micro-particles with free flow ability.
- Establish a simple and reproducible process.

Concerning the structure of this chapter first of all the three different basic techniques are described in detail. Afterwards the main influences on the properties of the micro-particle are presented. Therefore the commonly used compounds for the two phases, namely the disperse and continuous phase are summarized.

## 2.2. Solvent Evaporation and Extraction based Processes

According to Herrmann et al. [13] the most popular methods for the preparation of micro-particles from hydrophobic polymers are organic phase separation and solvent removal techniques. This method neither requires high temperatures nor phase separation-inducing agents. Nevertheless it is possible to achieve controlled particle sizes in the nano to micrometer range [9].

Freitas et al. [9] defined the four major steps of the micro-particle preparation by solvent extraction/evaporation as listed below.

1. Dispersion or dissolution of the API (Active Pharmaceutical Ingredient), in an organic solvent which contains the matrix forming material (for example PLGA polymer).
2. This organic phase in then emulsified in a second continuous (frequently aqueous) phase immiscible with the first one.
3. The solvent is extracted from the dispersed phase into the continuous phase, which is optionally accompanied by the evaporation of the solvent. As a consequence of that the droplets are transformed into solid micro-particles.
4. In this step the micro-particles are harvested and dried.

A schematic overview over these four major steps is shown in Figure 2-1Figure 1-1.

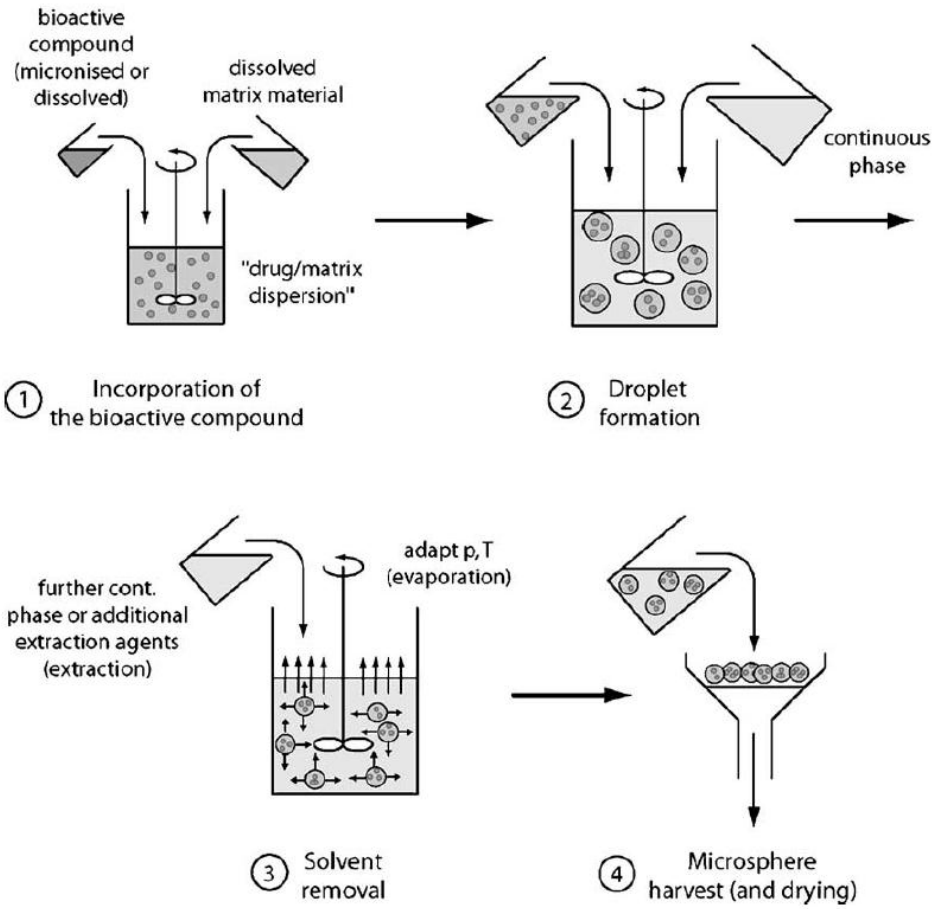



**Figure 2-1: Schematic overview over the four principal process steps in the micro-particle preparation by solvent extraction/evaporation [9].**

Classically, the drug-containing organic polymer solution is emulsified into an aqueous phase which normally contains a suitable stabilizer. The solubility properties of the drug determine if it is dissolved, dispersed or emulsified as an aqueous solution into the organic polymer solution [13].

## 2.3. Phase Separation (Coacervation)

According to IUPAC (International Union of Pure and Applied Chemistry), coacervation is defined as the separation of colloidal systems into two liquid phases [21]. This long established and widely used method for micro-encapsulation of biological materials relies upon a decrease of the polymer solubility by addition of a non-solvent [23] [11].

According to Nihant et al. [23], the point has to be reached where the two liquid phases are formed: a polymer-rich coacervate where the dissolved drug is entrapped and a supernatant liquid phase depleted in polymer.

In Figure 2-2 a schematic representation of the coacervation process is shown. As defined by Jyothi et al. [17], the major steps of a micro-encapsulation by coacervation process are:

1. Dispersion of the core material in a solution of shell polymer.
2. Separation of coacervate from solution.
3. Coating of the core material by the produced micro-droplets of coacervate.
4. Formation of a continuous shell around core particles trough the coalescence of coacervate.

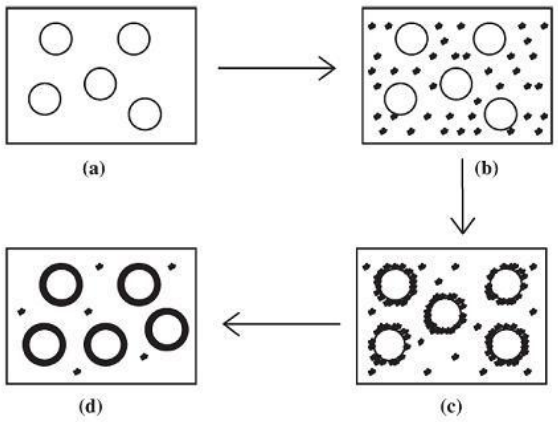



**Figure 2-2: Schematic representation of the coacervation process [17].**

According to Luan [20] the major disadvantages of coacervation method are the difficulties in scaling-up, the use of large amount of organic solvent and the high residual of solvent and coacervating agents in the micro-spheres. Also the production of micro-particles in the low micrometer size range is difficult [9].

## 2.4. Spray Drying

Spray drying has been used for many years as micro-encapsulation technique because of its simplicity, the high throughput and the low-costs [17] [33] [9]. In comparison to the previous mentioned methods, spray drying, is easily to be scaled up [20].

As shown in Figure 2-3, the core particles which contain the drug are dispersed in a polymer solution and atomized/sprayed into a hot chamber [17] [20]. This leads to an instantaneously evaporation of the solvent whereby relatively rapid solidification of the shell material onto the core particles is effected [33] [20].

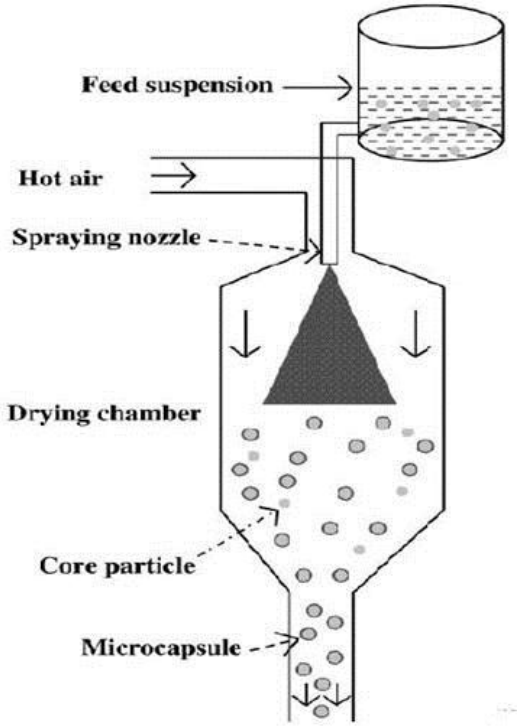



**Figure 2-3: Schematic illustrating the process of micro-encapsulation by spray-drying [17].**

The greatest disadvantages of this method are that it must not be used for highly temperature sensitive-compounds and that there are large amount of product loss during process, which result in a low yield [9]. Furthermore, spray drying is also prone to produce agglomeration. Hence the control of the particle size is difficult [20].

## 2.5. Choice of Material

According to Izumikawaa et al. [15] the properties of the used compounds and the operating parameters have a big impact on the physical properties of the obtained micro-particles. These main factors are summarized in Figure 2-4.

In these section an overview over the different common used compounds for the two phases, namely the disperse and continuous phase are presented. The detail description of the compounds used in our process can be found in section 3.2.

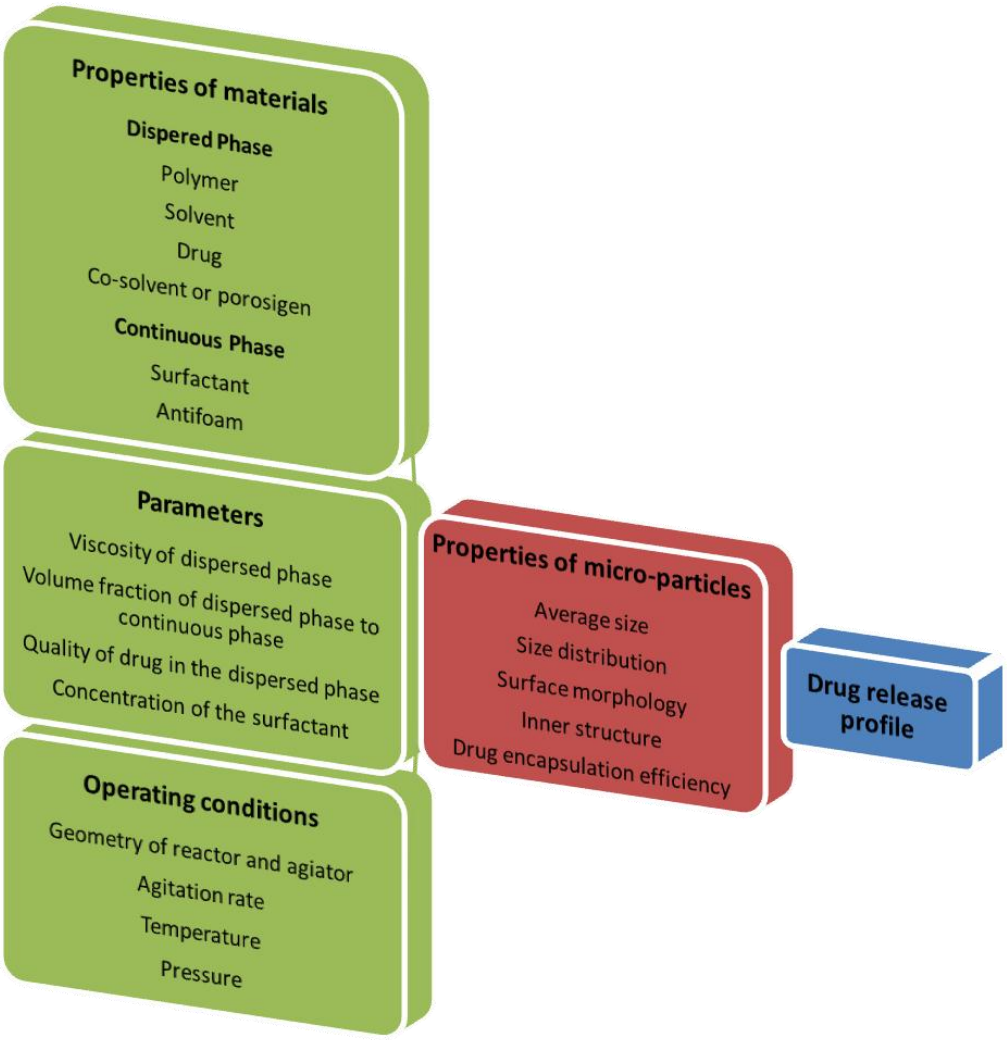



**Figure 2-4: Schema of the factors influencing the properties of micro-particles [19].**

### 2.5.1. Dispersed Phase

**Polymer**

An essential characteristic of the polymers used in the pharmaceutical industry are their biodegradability or biocompatibility. That means that the components should on the one hand degrade into harmless components which are either metabolized or excreted. On the other hand they should be physiologically acceptable and should not cause a negative local or systemic reaction after the administration.

Polymers and copolymer copolymers of lactic and glycolic acids are often used as the biomaterials for the microencapsulation of therapeutics because they provide the requested properties and the applications in humans are approved by the FDA (Food and Drug Administration) [19].

The desired drug release rate is determined by the physical properties of the polymer. So if one polymer cannot offer the wanted release characteristics a single polymer, called co-polymer can be synthesized from two different ones.

The main properties of the usually used polymers and copolymers for microencapsulation by solvent evaporation method are presented in Table 2-1.

**Table 2-1: Polymers commonly used for microencapsulation using solvent evaporation techniques [19].**

| Abbreviation | Complete name | Properties |
|---|---|---|
| PLGA, PLG | Poly(lactic-co-glycolic acid) or Poly(lactide-co-glycolide) | Good biodegradability and biocompatibility |
| PLA | Poly(lactic acid) or polylactide | Good biodegradability and biocompatibility, slow degradation rate compared to PLGA |
| PEG (used in co-polymer) | Poly(ethylene glycol) | Often synthesized with PLGA or with PLA to form a co-polymer with fast degradation rate |
| PHB | Poly-3-hydroxybutyrate | Bacterial storage polyester; slower degradation rate than polylactic polymers |
| PHB-HV | Poly-3-hydroxybutyrate with hydroxyvalerate | Bacterial storage polyester; slower degradation rate than polylactic polymers |
| EC | Ethyl cellulose | Degradable, biocompatible, approved by FDA for pharmaceutical application; low cost |
| PMMA | Polymethyl methacrylate | Non-degradable but biocompatible, approved by FDA; bone cement material; low cost; alternative polymer for scale-up investigation |

### Solvent

According to Li et al. [19] the solvent for the technique of microencapsulation by solvent evaporation, should fulfill the following criteria:

- being able to dissolve the chosen polymer;
- being poorly soluble in the continuous phase;
- having a high volatility and a low boiling point;
- having low toxicity.

The main solvents used in the literature and their properties are presented in Table 2-2.

**Table 2-2: List of solvents commonly used for microencapsulation by solvent evaporation [19].**

| Name | Advantages/Disadvantages |
|---|---|
| Chloroform | Low solubility in water; higher toxicity than dichloromethane |
| Dichloromethane (methylene chloride) | Dissolvation of most of the polymers; almost immiscible in water; high volatility and quite low boiling temperature; high toxicity |
| Ethyl acetate | Low toxicity/partially soluble in water; very low vapour pressure |
| Ethyl formate | Low toxicity; partially soluble in water |

## Alternative Components

In some cases a co-solvent and a porosity generator could be added to the disperse phase [19]. A co-solvent could be improve/enhance the dissolution of the drug and a porosity generator could increase the number of pores inside the micro-particle.

### 2.5.2. Continuous Phase

**Surfactant**

To reduce the surface tension of the continuous phase, avoid coalescence and agglomeration of the drops and, as well as to stabilize the emulsion frequently a surfactant, also called tensioactive agent, is employed. The most important criteria in selecting the surfactant are the desired size and the demand on the sphericity of micro-particles.

The amphiphilic behavior of the surfactants, that means one part of the molecule is hydrophilic and the other one is hydrophobic, causes the stabilizing effect by covering the surface of the drops.

According to Li et al. [19] there are four different surfactants classified by the nature of the hydrophilic part of the molecule:

- anionic,
- cationic,
- amphoteric
- and non-ionic

**Alternative Components**

In certain cases antifoam is added into the aqueous phase because the foaming problem disturbs the formation of the micro-particle [19].

# 3. Basic Engineering

## 3.1. Process Overview

The production of micro-particles represents an important unit operation in the pharmaceutical industry. The typical methods to manufacture solid particles from liquid solutions are described in detail in Chapter 2.

As illustrated in Figure 3-1, the investigated micro-particle formation process based on solvent extraction method can be separated into 5 stages:

1. Mixing Step (Droplet Formation)
2. First Extraction Stage (First Solvent Removal/Solidification Stage)
3. Intermediate Drying Stage
4. Second Extraction Stage (Second Solvent Removal/Solidification Stage)
5. Final Drying Stage

Concerning the structure of this chapter first of all the different used chemicals are presented. Then the different preparation steps of the three needed solution respectively the organic, the liquid and the continuous phase and the stages of the micro-particle process are describe in detail. After that an overview over the experimental set up is given and the most important details of the different units, namely the mixer, reactor and the cooling unit are shown.

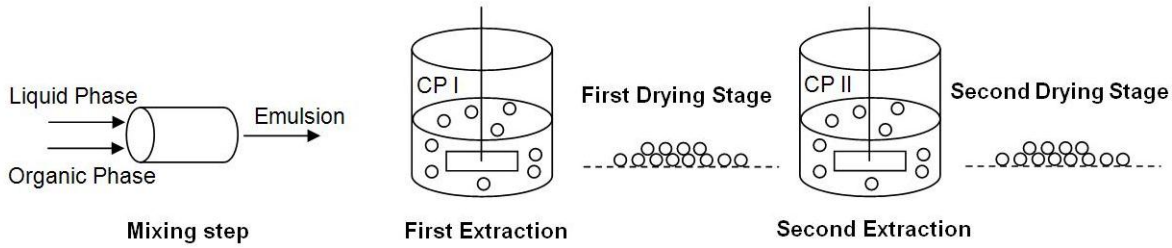



**Figure 3-1: Overview of the process.**

## 3.2. List of the Chemicals

The material parameters of the most important components are summarized in Table 6-3. The compounds used in our process are here described in detail.

### 3.2.1. Ethyl acetate

The often used solvent ethyl acetate respectively ethyl Ethanoate and commonly abbreviated EtOAc or EA is an organic compound with the formula $CH_3COOCH_2CH_3$. It shows promising potential as the solvent of the polymer (PLGA) in the micro-particle production based on solvent extraction method because its less toxic than the often used methylene chloride [19]. In comparison to methylene chloride, ethyl acetate is highly soluble. Hence, the micro-particles cannot form properly if the disperse phase is directly introduced in the continuous one. Due the sudden extraction of ethyl acetate the polymer precipitated into fibre-like agglomerates [10]. According to Li et al. [19] three methods can be used to resolve this problem:

1. The aqueous solution could be pre-saturated with solvent to slow down the extraction and prevent agglomerates.

2. The dispersed phase could be first emulsified in a little amount of continuous phase for the formation of the drops. After that this solution could be led into larger amount of continuous phase where the main part of the ethyl acetate is extracted.

3. The dispersed phase could again be emulsified in a little amount of continuous phase. This solution is then agitated and trough the evaporation of the solvent the solidification of the micro-particle occurs.

### 3.2.2. Benzyl alcohol

In the investigated process benzyl alcohol was used as the solvent of the API. It is an organic compound with the formula $C_6H_5CH_2OH$, a low toxicity, good solubility characteristics and a low vapor pressure.

### 3.2.3. Polyvinyl alcohol (PVA)

The polyvinyl alcohol is a protective colloid and was used to stabilize the emulsion droplets after the emulsification. It is a non-ionic surfactant which has excellent film forming, emulsifying, and adhesive properties [19].

### 3.2.4. Polymer (PLGA)

PLGA or poly (lactic-co-glycolic acid) is a copolymer and is most frequently used as the biomaterials for the microencapsulation of therapeutics [9]. It provides an excellent biocompatibility and biodegradability and the applications in humans are approved by the FDA (Food and Drug Administration) [19].

## 3.3. Preparation of the Solutions

The different required solutions respectively phases, as described in Chapter 2, had to be prepared separately.

The ingredients and the weighted sample of the three needed phases (the organic, the liquid and the continuous phase) are listed in Table 3-1 and Table 3-2. The different used chemicals are described in section 3.2.

The organic phase was prepared in two parts to prevent mutual interference between the different ingredients during the dissolution process of the PLGA and the API.

One consisted of the polymer (PLGA) and its solvent ethyl acetate, while the other one consisted of the API and its solvent benzyl alcohol. The first solution was stirred magnetically at 1200 rpm and the other one at 600 rpm for 15 minutes, respectively. After that they were combined and stirred together at 600 rpm for 2 minutes. To decompose the weight of the polymer to 80000 Dalton the organic phase was tempered for 3 hours in a water bath at 20 °C.

The liquid and the continuous phases were produced by mixing the different ingredients together. The liquid phase consisted of ethyl acetate and polyvinyl alcohol (PVA) and the ingredients of the continuous phase were water and ethyl acetate.

**Table 3-1: Ingredients of the first continuous phase.**

| Ethyl acetate | Water |
|---|---|
| [g] | [g] |
| 83,03 | 3440,00 |

**Table 3-2: Ingredients of the disperse phase.**

| Organic Phase | | | | Liquid Phase | |
|---|---|---|---|---|---|
| Polymer | API | Ethyl acetate | Benzyl alcohol | PVA | Ethyl acetate |
| [g] | [g] | [g] | [g] | [ml] | [g] |
| 8,80 | 6,13 | 44,76 | 18,90 | 635,00 | 45,00 |

### 3.3.1. Mixing Step

At the beginning the reactor was filled with the continuous phase and was cooled down to 5 °C. For the whole process the stirrer rate was set to 260 rpm.

In the mixing step the organic phase was emulsified into the liquid phase generating the initial stage of micro-particle shaping. Therefore, the two phases were pumped through a static mixer unit with two gear pumps. The different influences like speed and volume flow on the resulting emulsion droplet size and their distribution are presented in a publication from Kiss et al. [22].

### 3.3.2. First Extraction Stage

The produced emulsion was then led trough a custom build glass device in the reactor unit. There the two solvents (ethyl acetate and benzyl alcohol) were extracted into the surrounding aqueous continuous phase. As a consequence of the decrease of ethyl acetate concentration inside the micro-particles, which is the solvent of the polymer (PLGA), the emulsion droplets transformed into semi-solid micro-particles.

### 3.3.3. Intermediate Drying Stage

The first extraction step was assessed to be finished after 20 hours. In the intermediate drying stage the semi-solid particles were separated from the slurry. Afterwards the whole solution of the reactor was filled in a custom build vacuum-sieve-drying unit and the most of the aqueous phase was removed. The remaining micro-particles were dried with air for 24 hours at -5 °C.

### 3.3.4. Second Extraction Stage

After the intermediate drying stage the dried micro-particles were suspended again and a second extraction step with a different continuous phase was conducted. The temperature and the stirrer rate were held constant. The ingredients and weighted sample of the second continuous phase are listed in Table 3-3.

**Table 3-3: Ingredients of the second continuous phase.**

| Ethanol [ml] | Water [ml] |
|---|---|
| 1000,00 | 3000,00 |

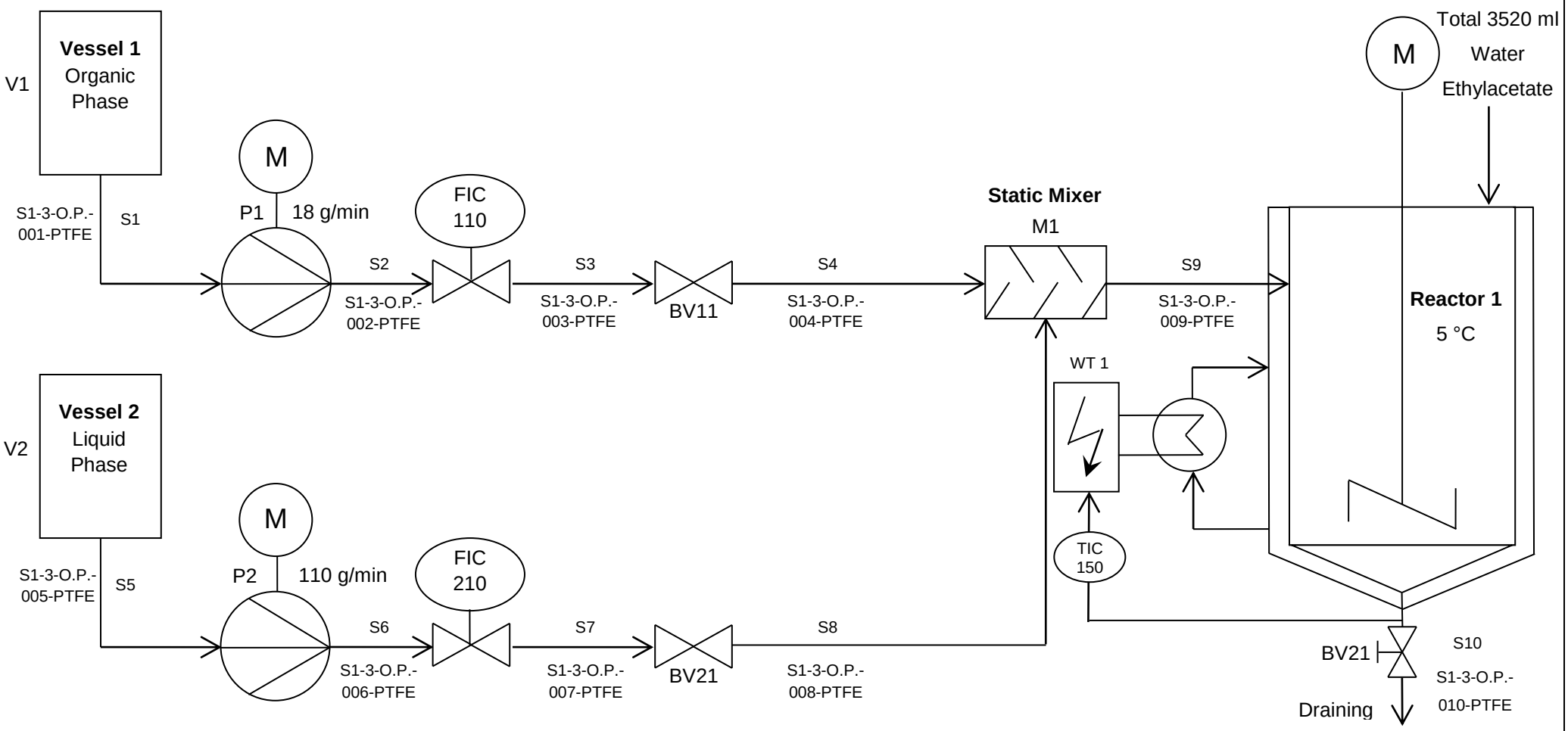
### 3.3.5. Final Drying Stage

At the end the micro-particles were dried again in the custom build vacuum-sieve-drying unit for about 5 days at -5 °C. When the water content and the residual solvent content of the micro-particles was under 0.2 mass percent this final drying process was over.

## 3.4. Overview over the Experimental Set up

For the detailed investigation of the process an experimental installation was build up in the laboratories of the Research Center Pharmaceutical Engineering (RCPE GmbH) in Graz. A picture of the final experimental installation is shown in Figure 3-2, while P&ID-Flow Diagram of the process is shown in Figure 3-3. The most important units are described in the following sub-sections.

**Figure 3-2: Experimental installation.**




| | | | |
|---|---|---|---|
| (Range of application) | (Adm. tol.) | Surface | Scale 1:1 / (Weight) |
| | | | (Basic material)<br>(Raw part number)<br>(Modell or casting number) |

| | Date | Name |
|---|---|---|
| Editor | 9.2.2010 | Pucher |
| Reviewer | | |
| Norm | | |

| Auth. | Revision | Date | Name |
|---|---|---|---|
| | | | |

# P&ID-Flow Diagram

Page

Pages

Source

Replacement for:

### 3.4.1. Mixer Unit

The emulsification of the organic and liquid phase is a very important step for the whole process. In fact, the final size of the micro-particles could be directly related to the size of the emulsion droplets. In our case the emulsions were produced by pumping the liquid and the organic phase with two Ismatec MCP-Z IP 65 gear pumps trough a custom build static mixer unit. After this static mixer unit the produced emulsion droplets were led into the reactor unit. In Figure 3-4 the mixer unit is shown. The most important facts about the gear pumps and there pump heads are summarized in Table 3-4, Table 3-5 and Table 3-6.

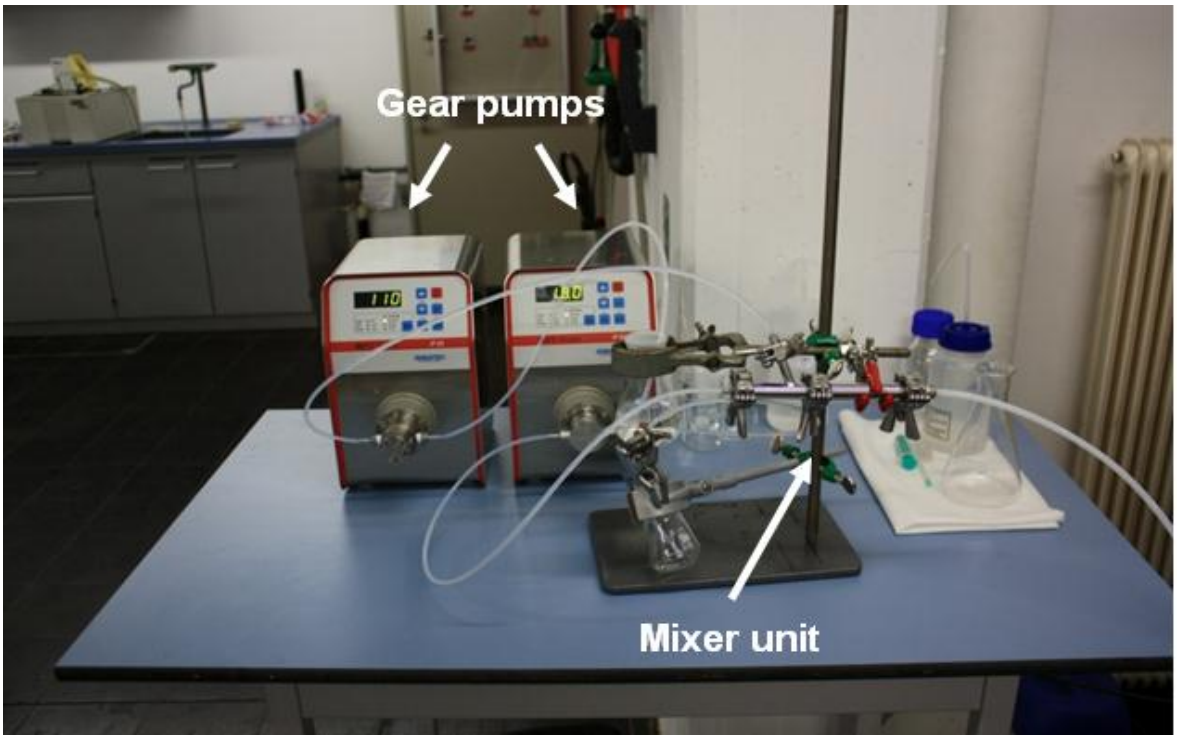



**Figure 3-4: Mixer unit.**

**Table 3-4: Data sheet of the gear pumps (MCP-Z IP 65).**

**ISMATEC MCP-Z IP 65**

| | | |
|---|---|---|
| **Flow rates**[1] | 1 – 7020 | [ml/min] |
| **Motor speed** | 60 – 6000 | [rpm] |
| **Speed setting** | digital speed setting | (resolution 1 rpm) |
| **Calibrating functions** | - for flow rate [ml/min]<br>- for dispensing volume [ml] | |
| **Power consumption** | 150 | [W] |
| **Weigth** | 6,9 | [kg] |

**Table 3-5: Data sheet of the gear pump head (Z-120).**

**ISMATEC Magnetically coupled Cavity Style pump head Z-120**

| | | |
|---|---|---|
| **Flow rates** | 38 – 3840 | [ml/min] |
| **Differential pressure**[2] | 3,5 / 50,8 | [bar/psi] |
| **System pressure, max** | 21 / 304,6 | [bar/psi] |
| **Operating temperature**[3] | -46 ... +54 | [°C] |
| **Gear Material** | PTFE | |
| **Seals** | PTFE | |
| **Stainless steel housing** | SS316 | |

[1] depending on mounted pump-head
[2] Flow rates without differential pressure
[3] With other seals up to 99°C possible

**Table 3-6: Data sheet of the gear pump head (Z-186).**

**ISMATEC Magnetically coupled Suction shoe style pump head Z-186**

| | | |
|---|---|---|
| **Flow rates** | 1 – 102 | [ml/min] |
| **Differential pressure** | 1,4 / 20,3 | [bar/psi] |
| **System pressure, max** | 21 / 305 | [bar/psi] |
| **Operating temperature**[4] | -46 ... +177°C | [°C] |
| **Gear Material** | Graphite | |
| **Seals** | PTFE | |
| **Stainless steel housing** | SS316 | |

The custom build static mixer unit consisted of ten mixer elements (SMX elements) build from Sulzer Chemtech AG and a custom build housing from Bamberger & CO. The mixer elements were cast of stellite and were placed into a camber in the housing. The different parts of the housing were then locked together with four TRI-Clamps.

The custom build housing was made of austenitic stainless steel 1.4404. In Figure 3-5 one SMX element is shown.

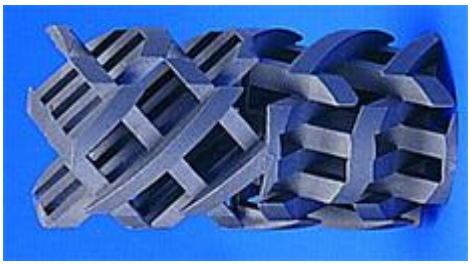


**Figure 3-5: Mixer elements.**

[4] With other seals up to 99°C possible

### 3.4.2. Reactor Unit

As described before, the produced emulsion droplets were led into the 5 liter double jacket lab reactor trough a custom build glass device. In Figure 3-6 the complete set up of the reactor unit is shown.

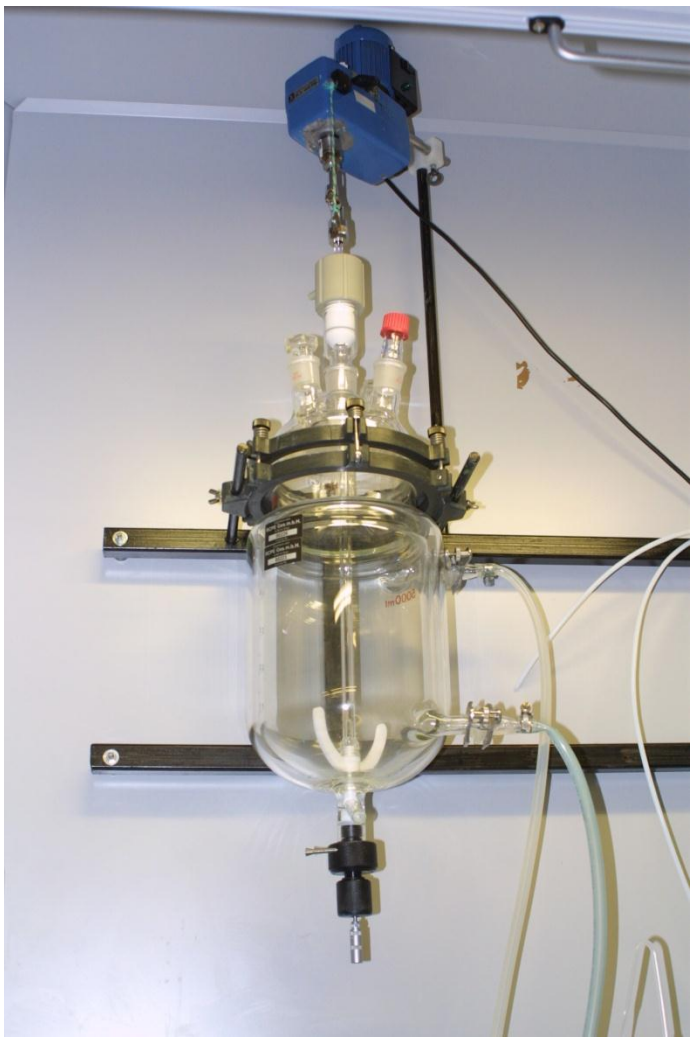


**Figure 3-6: Reactor unit.**

This lab reactor was made out of glass by Ruprechter Glasbläserei-Laborbedarf e.U. and its engineering drawing can be found in the Appendix.

The reactor was stirred by an impeller build by Schmizo AG. The shaft of the stirrer was made out of glass and the blade consisted of PTFE. The engineering drawing of the stirrer unit can be found in the Appendix as well.

### 3.4.3. Cooling Unit

During the extraction process the liquids in the reactor unit had to be tempered constantly under 5 °C in order not to excess the glass transition temperature of the polymer. This was achieved by a refrigerated/heating circulator (F 25 ME), shown in Figure 3-7, build by Julabo Labortechnik GmbH. This circulator was connected with silicon tubes to the double jacket lab reactor. To provide the needed temperature, Glycol was used as cooling liquid. In Table 3-7 the most important parameters about the circulator are summarized.

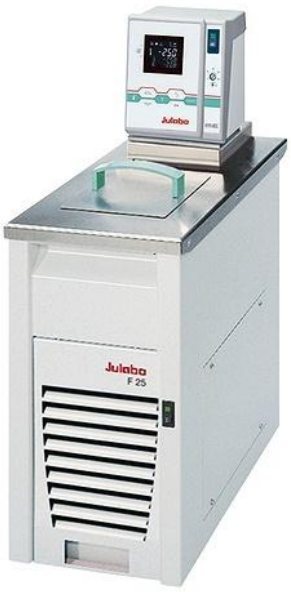



**Figure 3-7: Cooling unit.**

**Table 3-7: Data sheet of the circulator.**

| **Julabo F 25 ME** | | | | |
|---|---|---|---|---|
| **Working temperature range** | -28 ... +200 | | [°C] | |
| **Temperature stability** | 0,01 | | [°C] | |
| **Heater capacity** | 2000 | | [W] | |
| **Filling volume** | 4,5 | | [liters] | |
| **Ambient temperature** | 5 ... +40 | | [°C] | |
| **Cooling capacity** | 20 | 0 | -20 | [°C] |
| | 350 | 250 | 60 | [W] |

# 4. Experimental Work

## 4.1. Introduction

The main target of the experimental work was to monitor the micro-particle size during the extraction process. Therefore, two measurements systems from Sympatec GmbH, namely HELOS, a laser diffraction system, and QICPIC, an image analysis system, were used. These two systems are described in section 4.2.

Basically, experimental data were produced, in cooperation with M. Sc. Nikolett Kiss, in order to validate the numerical model developed in Chapter 5. This validation is presented in sub section 6.5.

## 4.2. Particle Size Analysis

### 4.2.1. HELOS

The HELOS (Helium-Neon Laser Optical System), shown in Figure 4-1, is a laser diffraction sensor from Sympatec GmbH. This system uses one measuring principle (laser diffraction in the parallel laser beam) for the its whole measuring range from 0.1 µm to 8750 µm.


**Figure 4-1: HELOS/BR and wet disperser CUVETTE 50ML/US for small sample quantities [37].**

The supplied software, WINDOX, offers two different evaluation modes, which allows the analysis of extremely wide size distributions at highest precision:

- FREE: a parameter free solution basing on Fraunhofer diffraction
- MIEE: basing on precision Mie theory extended to the full size range

The classical field of application is the particle size analysis of dry and wet samples, i.e. of powders, suspensions, emulsions or sprays and the absolute accuracy is typically within ±1% with respect to the standard metre.

The HELOS analysis system combines high resolution and guaranteed reproducibility with high speed data acquisition.

### 4.2.2. QICPIC

The QICPIC, shown in Figure 4-2, is an image analysis system from Sympatec GmbH which combines particle size with shape analysis for particles with a speed of up to 100 m/s. For dry particles, a dispersion unit guarantees proper dispersion of agglomerated fine and cohesive powders. In addition it is also possible to apply wet disperser as LIXELL and SUCELL for the analysis of emulsions and suspensions.

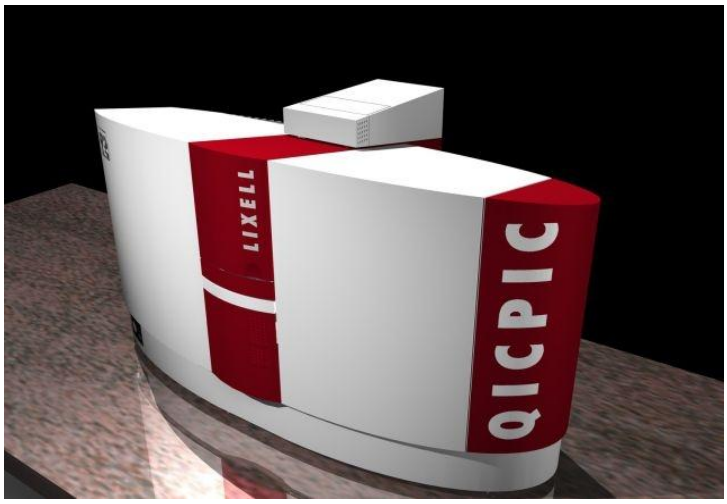



**Figure 4-2: Particle size analysis of suspensions and emulsion with the wet disperser LIXELL inserted in the measuring zone of a QICPIC [38].**

The QICPIC analysis system is able to measure particles between 1 µm and 20 mm and the high performance data compression allows grabbing up to 500 images per second. These leads to a high statistical security of measurement results in short analysis time

Besides that the supplied software, WINDOX, is capable of simultaneous calculate different size and shape features.

In addition QICPIC fulfills the security requirements of the FDA and is compliant to 21 CFR Rule 11.

## 4.3. Micro-Particle Sizing during the Extraction Process

The size distribution of the micro-particles, during the extraction process, was determined using the HELOS laser diffraction instrument. At different instants of time, samples were taken out of the extraction reactor and the micro-particle size was determined right away to avoid aging influences. Therefore the samples were filled into a cuvette provided by Sympatec GmbH and measured in triplicate. The Fourier optic R5, which has a drop size measuring range from 0,5/4,5 to 875 µm, was here used. The evaluations of the different diffraction patterns were evaluated with the FREE (Fraunhofer Enhanced Evaluation) mode which is provided from the evaluation software.

**Median value $x_{50}$:**

The median value is defined as the particle diameter that corresponds to a cumulative distribution $Q_3$ of 50 %. This value divides the density distribution $q_3$ into two equal halves.

**Sauter mean diameter SMD/$d_{32}$:**

The Sauter mean diameter SMD is defined as the diameter of that spherical particle which has exactly the same volume/surface area ratio as a particle of interest. It's commonly in literature defined as:

$$d_{32} = \frac{\sum_i N_i d_i^3}{\sum_i N_i d_i^2} \tag{4-1}$$

Here, $N_i$ is the number and $d_i$ is the diameter of the particles of interest.

# 5. Modeling

## 5.1. Introduction and MATLAB Overview

The final quality of the considered micro-particles can be predicted and controlled only by a deep understanding of each step of the production process, namely mixing, hardening/extraction and drying.
As described in this section, the modeling of the diffusion process was one of the main challenges of this work. The developed numerical method was based on the MATLAB software from MathWorks Inc. The MATLAB environment and its solvers are commonly used for technical computing and data visualization in many engineering fields.

Concerning the structure of this chapter first of all the developed mass transport model is described in detail. Then the estimation procedure of the mass transport coefficient and the diffusion coefficient are declared. Afterwards an overview over the MATLAB-PDEPE solver is given. Then the analytical verification of the model without shrinking is presented. Finally the method to numerically compute the diameter reduction respectively the shrinking model is presented.

All the different code parts concerning the numeric model can be found in the appendix.

## 5.2. Diffusion Model

### 5.2.1. Mass Transport Model

As described in chapter 3, the extraction process is a process with four different components:

1. Solvent (Ethylacetate)
2. Co-Solvent (Benzylalcohol)
3. Polymer
4. API

Basically, the mass transfer in the micro-particle can be assumed to be a multicomponent diffusional process. The chosen approach to resolve this physical problem is based on the work of Li et al. [18].

Here, the continuity equation for multiple component diffusion for an infinitesimal volume can be written as follows:

$$\frac{\partial \varphi_i}{\partial t} = -\nabla \cdot (J_i + \varphi_i v_3) \qquad (5\text{-}2)$$

where i represent the four different components in the disperse phase, $\varphi_i$ is the volume fraction, $J_i$ is the volume flux relative to the polymer velocity and $v_3$ is the velocity of the polymer.

For a first analysis of the problem the term $v_3$ was assumed to be equal to zero, meaning that the shrinking of the particle was considered negligible. For the development of the mathematical model the following assumptions have been made:

- The particles were assumed to be spherical and uniformly distributed in $\varphi$ and $\theta$.
- The porosity of the micro-particles was neglected.
- It was assumed that there were no interactions between the different components which mean that the different components can diffuse independently from each other.

The volume flux is defined as follows:

$$J_i = \frac{1}{r}\frac{D_i}{RT}\nabla_{r_n}\mu_i \tag{5-3}$$

where $D_i$ is the concentration-dependent binary diffusion coefficient, R is the gas constant, T is the temperature and $\mu_i$ is the chemical potential for each component i.

The normalized position variable of the micro-particle $r_n$ is defined as follows:

$$r_n = \frac{r}{r_{max}} \tag{5-4}$$

Here r is the position vector and $r_{max}$ is the radius of the micro-particle.
With the presented assumptions and the equation found by Bird et al. [4], the equation can be transformed in:

$$\frac{\partial\varphi_i}{\partial t} = \frac{1}{r^2}\cdot\frac{\partial}{\partial r}\cdot\left(\frac{r^2}{R\cdot T}\cdot D_i\cdot\frac{\partial\mu_i}{\partial r}\right) \tag{5-5}$$

The chemical potential for this system can be expressed as a function of different interaction parameters and of the volume fraction of the different components. As described before a four component system was assumed. Since the API loading was generally low the polymer and API were considered as one component. According to Tompa et al. [31] and Flory et al. [8], the chemical potential for the solvent i is defined as:

$$\begin{aligned}\frac{\Delta\mu_i}{RT} &= \ln(\varphi_1) - \alpha\varphi_2 - \beta\varphi_3 - \gamma\varphi_4 \\ &+ (1 + \chi_{12}\varphi_2 + \chi_{13}\varphi_3 + \chi_{14}\varphi_4)(1 - \varphi_1) - \alpha\chi_{23}\varphi_2\varphi_3 \\ &- \gamma\chi_{34}\varphi_3\varphi_4 - \beta\chi_{24}\varphi_2\varphi_3\end{aligned} \tag{5-6}$$

where α, β, γ are the ratios of the molar volumes $\bar{v}_i$ which can obtain from the Eqn. (5-7) and $\chi_{ij}$ represent the binary interaction parameter.

$$\alpha = \frac{\bar{v}_1}{\bar{v}_2}, \beta = \frac{\bar{v}_1}{\bar{v}_3}, \gamma = \frac{\bar{v}_1}{\bar{v}_4} \tag{5-7}$$

The molar volume for the component i is defined as follows:

$$\bar{v}_i = \frac{M_i}{\rho_i} \tag{5-8}$$

where $M_i$ is the molar mass and $\rho_i$ is the mass density.
The presented Eqn. (5-6) represents the chemical potential for a quaternary system and can be simplified during the formation stage into a ternary system which can be rewritten as:

$$\frac{\Delta\mu_i}{RT} = \ln(\varphi_1) - \alpha\varphi_2 - \beta\varphi_3 + (1 + \chi_{12}\varphi_2 + \chi_{13}\varphi_3) - \alpha\chi_{23}\varphi_2\varphi_3 \tag{5-9}$$

And for the solvent removal stage the chemical potential is defined as:

$$\frac{\Delta\mu_i}{RT} = \ln(\varphi_1) - \beta\varphi_3 - \gamma\varphi_4 + (1 + \chi_{13}\varphi_3 + \chi_{14}\varphi_4)(1 - \varphi_1) - \gamma\chi_{34}\varphi_3\varphi_4 \tag{5-10}$$

The Eqn. (5-9) and Eqn. (5-10) are the final expressions of the chemical potential for a ternary system as described in literature (see e.g. Li et al. [18]).

The binary interaction parameter $\chi_{ij}$, the gas constant R, the process temperature T and also the ratios of the molar volumes α, β, γ remain constant throughout the entire process. These constants may also be neglected for the modeling.

The final differential equation for the mass transfer in the micro-particle is shown in (5-11):

$$\frac{\partial\varphi_i}{\partial t} = \frac{1}{r^2} \cdot \frac{\partial}{\partial r} \cdot \left( r^2 \cdot D_i \cdot \frac{\partial\varphi_i}{\partial r} \right) \tag{5-11}$$

The terms on the left hand side are they so-called accumulation terms, while the terms on the right hand side are the so-called diffusion terms.

### 5.2.2. Initial and Boundary Conditions

In order to solve the above-presented differential equation, the MATLAB PDEPE solver was used (please see section 5.3 for more details about this module).
The initial condition for the volume fraction $\varphi_i$ of the component i is defined as:

$$\varphi_{i(r,t=0)} = \varphi_i{}^0 \quad i = 1,2,4 \tag{5-12}$$

At the beginning (t=0) all components were equally distributed and the volume fraction was determined by the initial composition of the different components.
The first boundary condition for the volume flux of the component i is:

$$J_{i(0,t)} = 0 \tag{5-13}$$

This boundary applied no volume flow at the center of the micro-particle and is also called regularity condition.
The second boundary condition describes the mass flow from the interface of the micro-particle into the continuous phase:

$$-D_i \cdot \frac{\partial \varphi_{i,r=R}}{\partial r} = \beta_{i,t} \cdot \Delta\varphi \tag{5-14}$$

Here, $D_i$ is the binary diffusion coefficient, $\beta_{i,t}$ is the mass transport coefficient and $\Delta\varphi$ is the volume fraction difference between the disperse and the continuous phase.
According to the formulation of the film theory [2], the mass transfer between particle (disperse phase) and extraction medium (continuous phase) depends on the mass transfer coefficient and the volume fraction difference between the interface and the continuous phase, namely $\varphi_{DP,i,\infty}$ and $\varphi_{CP,i,\infty}$.

Generally, the transport process can be divided into three components:

- Transport of component i from the disperse phase to the interface;
- Transport of component i through the interface;
- Transport of component i from the interface into the continuous phase.

A schematic representation of the volume fraction profiles of component i at the interface between disperse and continuous phases is shown in Figure 5-1.

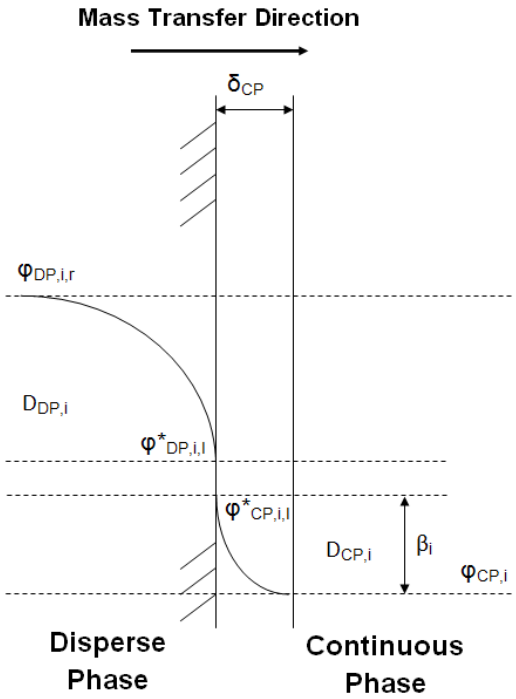



**Figure 5-1: Schematic representation of the volume fraction profiles of component $\varphi_i$ at the interface.**

| | |
|---|---|
| $\varphi_{i,DP,\infty}$ | Volume fraction in the DP |
| $\varphi_{i,CP,\infty}$ | Volume fraction in the CP |
| $\varphi^*_{i,DP,I}$; $\varphi^*_{i,CP,I}$ | Equilibrium volume fraction at the interface |
| $\beta_i$ | Mass transfer coefficient |
| $D_{i,DP}$; $D_{i,CP}$ | Diffusion coefficient in the DP and CP |
| $\delta_{CP}$ | Interface thickness |

The transport through the interface is usually assumed to be unresisting. Therefore, the thickness of the interface is regarded to be infinitely thin. Another assumption is that phase equilibrium is present at the interface.

Generally, the mass transport for a liquid-liquid system can be defined by Eqn. (5-15) to Eqn. (5-18).

$$\dot{n}_{i,DP} = \beta_{i,DP} \cdot (\varphi_{i,DP,\infty} - \varphi^*_{i,DP,I}) \tag{5-15}$$

$$\dot{n}_{i,CP} = \beta_{i,CP} \cdot (\varphi^*_{i,CP,I} - \varphi_{i,CP,\infty}) \tag{5-16}$$

$$\dot{n}_i = \dot{n}_{i,CP} = \dot{n}_{i,DP} \tag{5-17}$$

$$\dot{n}_i = \frac{1}{\frac{1}{\beta_{i,DP}} + \frac{1}{K \cdot \beta_{i,CP}}} \cdot \left(\varphi_{i,DP,\infty} - \frac{1}{K} \cdot \varphi_{i,CP,\infty}\right) \tag{5-18}$$

Here, $\dot{n}_i$ is the molar flow, $\beta_{i,j}$ and $\varphi_{i,j,\infty}$ are the mass transport coefficient and the volume fraction of the component i in the phase j, respectively, while K is the equilibrium constant, also known as Nernst distribution coefficient.

In our approach, the mass transport in the disperse phase was modeled through a multicomponent diffusion model. The exact algorithm to determine the value of the mass transport coefficient $\beta_{i,CP}$ for the continuous phase is presented in section 5.2.3. Furthermore, this coefficient depends on the diameter of the micro-particle and is a time-dependent value.

The equilibrium constant K is defined as follows:

$$K = \frac{\varphi^*_{CP,i,I}}{\varphi^*_{DP,i,I}} \gg 1 \tag{5-19}$$

The value of K becomes strongly greater than 1 concerning the big volume difference between the continuous and disperse phase on the one hand and the fact that the saturation concentration for the solvents in the continuous phase is definitely higher than for the disperse phase on the other hand.

This means that the concentration of the two solvents in the continuous phase could be negligible and set equal to zero for the entire process. As a consequence of that assumption the extracted components diffuse very fast from the interface into the continuous phase.

### 5.2.3. Estimation of the Mass Transport Coefficient

The mass transport rate from the interface of the micro-particles into the continuous phase basically depends on the mass transport coefficient $\beta_i$ and the ambient concentration $\varphi_{i,CP,\infty}$. Moreover, the value of $\beta_i$ is highly dependent on the surrounding flow regime, on the diameter of the micro-particle $d_{P,t}$ and on the binary diffusion coefficient $D_i$ of the solvent i in the continuous phase. The mass transfer coefficient is usually expressed as a function of the Sherwood number Sh, the binary diffusion coefficient $D_i$ and the diameter of the micro-particle $d_{p,t}$ as follows:

$$\mathrm{Sh} = \frac{\text{convective mass transfer coefficent}}{\text{diffusive mass transfer coefficent}} = \frac{\beta_i \cdot d_{P,t}}{D_i} \qquad (5\text{-}20)$$

$$\beta_i = \frac{D_i \cdot \mathrm{Sh}}{d_{P,t}}$$

The dimensionless Sherwood number represents the ratio of convective to diffusive mass transport and can be expressed as function of Reynolds and Schmidt numbers.

$$\mathrm{Sh} = f(\mathrm{Re}, \mathrm{Sc}) \qquad (5\text{-}21)$$

The dimensionless Reynolds number represents the ratio of inertial forces to viscous forces and is used to determine whether a flow will be laminar or turbulent.

$$\mathrm{Re} = \frac{\text{inertial forces}}{\text{viscous forces}} = \frac{v \cdot d_{P,t}}{\gamma_{CP}} \qquad (5\text{-}22)$$

Here, v is mean velocity of the object relative to the fluid and $\gamma_{CP}$ is the kinematic viscosity of the continuous phase.

As presented above, the mass transport coefficient depends on the diameter of the micro-particle. This means that in a shrinking process its value changes over time.

For our case, the mean fluid velocity in the extraction tank was estimated with a correlation found by Kolmogoroff for isotropic turbulence:

$$v = v_0 \cdot \left(\frac{d_{P,t}}{d_R}\right)^{\frac{1}{3}} = \pi \cdot n \cdot d_R \cdot \left(\frac{d_{P,t}}{d_R}\right)^{\frac{1}{3}} \tag{5-23}$$

where n is the revolution rate of the stirrer and $d_R$ is the diameter of the stirrer blade.

The dimensionless Schmidt number Sc is the ratio of momentum diffusivity (viscosity) and mass diffusivity, expressed in terms of kinematic viscosity $\gamma_{CP}$ and binary diffusion coefficient $D_i$.

$$Sc = \frac{\text{viscous diffusion rate}}{\text{mass diffusion rate}} = \frac{\gamma_{CP}}{D_i} \tag{5-24}$$

The Schmidt number is used to characterize fluid flows in which momentum and mass diffusion-convection processes happen simultaneously.

There are many empirical and semi-empirical correlations to express the dimensionless Sherwood number as a function of the Reynolds and Schmidt numbers shown in equations Eqn. (5-22) and Eqn. (5-24). The correlations listed in Table 5-1 were found by different researchers and are described in a book from Tosun [32]. All these correlations are only valid for a certain range of Reynolds and the Schmidt number.

**Table 5-1: Empirical and semi empirical Sherwood correlations.**

| | **Reynolds number** | **Schmidt number** | **Sherwood number** |
|---|---|---|---|
| | **Re** | **Sc** | **Sh** |
| **Garner/Suckling** | 100 ÷ 700 | 1100 ÷ 2200 | $2 + 0{,}95 \cdot \sqrt{Re} \cdot Sc^{\frac{1}{3}}$ |
| **Frössling** | > 100 | ≤ 1000 | $2 + 0{,}552 \cdot \sqrt{Re} \cdot Sc^{\frac{1}{3}}$ |
| **Steinberger/Treybal** | 10 ÷ 17000 | 1 ÷ 70000 | $2 + 0{,}347 \cdot Re^{0{,}62} \cdot Sc^{0{,}31}$ |
| **Rowe et al.** | 25 ÷ 1150 | 1220 | $0{,}79 \cdot \sqrt{Re} \cdot Sc^{\frac{1}{3}}$ |

As shown in Table 5-2, the values from Reynolds and Schmidt numbers at the beginning fit the correlation found by Steinberger and Treybal exactly, namely Reynolds number

from 10 to 17000 and a Schmidt number from 1 to 70000. Therefore, this correlation was used for further calculations.

**Table 5-2: Values of Reynolds and Schmidt number.**

| | Reynolds number | Schmidt number | Sherwood number |
|---|---|---|---|
| | **Re** | **Sc** | **Sh** |
| **Ethylacetate** | 18,628 | 8,680 | 6,160 |
| **Benzylalcohol** | | 8,980 | 6,200 |

### 5.2.4. Diffusion Coefficient

As mentioned in section 5.1, the differential equations were solved with MATLAB. The different code parts concerning the calculation of the value of the binary diffusion coefficient can be found in the appendix. At the beginning of the extraction process, the binary diffusion coefficient could be estimated with an equation from Hayduk and Minhas [30]:

$$D_i = 1{,}25 \cdot 10^{-8} \cdot (V^{-0{,}19} - 0{,}292) \cdot T^{1{,}52} \cdot \eta^{\epsilon_b} \qquad (5\text{-}25)$$

where V is the molar volume of solute of solute at normal boiling point, T is the temperature and η is the dynamic viscosity. The term $\epsilon_b$ is defined as:

$$\epsilon_b = \frac{9{,}58}{V} - 1{,}12 \qquad (5\text{-}26)$$

This equation is valid for liquid-liquid system and represents a good approach to find initial values for the binary diffusions coefficient for each solvent.

Another correlation for the binary diffusion coefficient was found by Wilke and Chang [36]. This equation is also valid for diffusion in water and in non-associated solvents.

$$D_i = 7{,}4 \cdot 10^{-8} \cdot \frac{(x \cdot M)^{\frac{1}{2}} \cdot T}{\eta \cdot V^{0{,}6}} \qquad (5\text{-}27)$$

Here, M is the molecular weight of solvent and x is an association parameter.

The following paragraph is cited from their paper [36].

> "The correlation represented by equation 19 is satisfactory for estimation of diffusion coefficient in dilute solution with sufficient precision for most engineering purposes, i.e., about 10% average error[5]. It must be emphasized that the diffusion process is extremely complex and that any rigorous treatment must consider solute-solvent interaction in a more detailed manner that the present

[5] For 285 points among 251 solute-solvent systems of this study

relation could possibly imply[6]. Although the present functional relationship of diffusion coefficient to solute molar volume rests upon some qualitative theoretical foundation, the relationship to solvent molecular weight is strictly empirical."

Both above described correlations provide different starting values for the binary diffusion coefficient. The correlation found by Wilke and Chang [36] is typically used in literature (see e.g. Wang et al. [35]), so it was also used in our work.

As described before, the extraction of the solvents leads to the hardening of the micro-particle. Furthermore, with the increase of the volume fraction of the polymer the value of the diffusion coefficient decreases. This means that the diffusion coefficient is highly depending on the polymer volume fraction in the micro-particle. The correlation between the diffusion coefficient and the volume fraction of the solvent was found by Reuvers et al. [25] and used in the work by Li et al [18] and can be written as:

$$D_i = r_{1,i} \cdot 10^{-(r_{2,i} + r_{3,i} \cdot \varphi_P)} \qquad (5\text{-}28)$$

Where the parameters $r_{1,i}$, $r_{2,i}$ and $r_{3,i}$ are system dependent constants and $\varphi_P$ is the volume fraction of the polymer.

[6] Diffusion of iodine in aromatic hydrocarbons, for example, has been excluded from the present correlation because of known complex formation.

## 5.3. PDEPE Solver Description

In order to solve the one-dimensional parabolic-elliptic partial differential equations (PDE) described in the previous sections, the MATLAB-PDEPE solver was used and the system of equations was implemented in MATLAB 7.9.0 R2009b. All the different code parts can be found in the appendix.

In the following chapter an overview of the numerical solution method is given. The following description is a summary/abstract of several instructions which can be found in the books "MATLAB kompakt" by Schweizer [29] and "MATLAB guide" by Desmond et al. [14], as well as in the MATLAB product help [1].

Basically, the MATLAB's PDEPE solves a class of parabolic/elliptic PDE systems. "These systems involve a vector-valued unknown function u that depends on a scalar space variable, x, and a scalar time variable, t." [14]
The PDEPE algorithm is based on the discretization of the space variable, xmesh, and uses a second-order spatial discretization method based on the xmesh values. This means that the choice of xmesh has a strong influence on cost and accuracy of the numerical solution.

The generated system of ordinary differential equations is then solved by the routine "ode15s". This routine is a variable order solver based on the numerical differentiation formulas (NDFs), which is capable of solving stiff DAE systems.

Generally, the PDEPE module in MATLAB solves PDEs of the form:

$$c\cdot\left(x,t,u,\frac{\partial u}{\partial x}\right)\cdot\frac{\partial u}{\partial t}=x^{-m}\cdot\frac{\partial}{\partial x}\cdot\left(x^{m}\cdot f\left(x,t,u,\frac{\partial u}{\partial x}\right)\right)+s\cdot\left(x,t,u,\frac{\partial u}{\partial x}\right) \qquad (5\text{-}29)$$

The solution function $u_{(x,t)}$ is limited to the finite space interval $a \le x \le b$ and the time interval $t_0 \le t \le t_f$. The exponent m can be 0, 1, or 2, corresponding to slab, cylindrical, or spherical symmetry, respectively. The function c is a diagonal matrix and the flux and source function f and s are vector valued.

For $a \le x \le b$, thus all positions x of the discretized domain, and $t = t_0$the solution components satisfy initial conditions of the form:

$$u_{(x,t_0)} = u_{0(x)} \tag{5-30}$$

For x=a and all times t ($t_0 \leq t \leq t_f$) the solution must satisfy Eqn. (3-31) for particular functions $p_a$ and $q_a$:

$$p_{a(x,t,u)} + q_{a(x,t)} \cdot f\left(x, t, u, \frac{\partial u}{\partial x}\right) = 0 \tag{5-31}$$

Similarly, for x=b and all times t ($t_0 \leq t \leq t_f$),

$$p_{b(x,t,u)} + q_{b(x,t)} \cdot f\left(x, t, u, \frac{\partial u}{\partial x}\right) = 0 \tag{5-32}$$

must hold for particular functions $p_b$ and $q_b$. Elements of $q_a$ and $q_b$ are either identically zero or never zero.

A call to PDEPE has the general form

$$\text{sol} = \text{pdepe}(\text{m}, @\text{pdefun}, @\text{pdeic}, @\text{pdebc}, \text{xmesh}, \text{tspan}, \text{options})$$

The input parameter m corresponds to the above-discussed symmetry parameter and can take the values 0, 1 and 2. The vector xmesh defines the x values at which the numerical solution is computed. The vector tspan specifies the time points where the solution is to be returned.

“The time integration in pdepe is performed by “ode15s” and the actual timestep values are chosen dynamically-the tspan points simply determine where the solution is returned and have a little impact on computational cost or accuracy. The default properties of the “ode15s” can be overridden via the optional input argument options.

The output argument sol is a three-dimensional array such that sol(j,k,i) is the approximation to the i-th component of u at the point t=tspan(j) and x=xmesh(k).” [14]

The function pdefun has the form:

$$\text{function } [\text{c}, \text{f}, \text{s}] = \text{pdefun}(\text{x}, \text{t}, \text{u}, \text{DuDx})$$

and accepts the space and time variables together with the vectors u and DuDx, which approximate the solution u and the partial derivate $\partial u \partial x$. “The function returns vectors containing the diagonal of the matrix c, as well as the flux and source functions f and s.
Initial conditions are encoded in the function pdeic, which takes the form:

$$\text{function u0} = \text{pdeic(x)}$$

Finally, the function pdebc, defined as

$$\text{function [pa, qa, pb, qb]} = \text{pdebc(xa, ua, xb, ub, t, p1, p2, \ldots.)}$$

evaluates $p_a$, $q_a$, $p_b$ and $q_b$ for the boundary conditions at xa=a and xb=b.
The remaining input arguments p1, p2, … are optional problem parameters that are passed to the functions pdefun, pdeic and pdebc.” [14]

## 5.4. Analytical Verification without Shrinking

To prove the solution of Eqn. (5-11) with the boundary conditions in Eqn. (5-13) and Eqn. (5-14), the numerical results were compared with an analytical solution of a similar system. Precisely, the solution for surface evaporation found by John Crank [7] was used as an analytical benchmark for the discretized method used in this work. To compare the above presented numerical algorithm with the analytic one, the same parameters must be used. The assumption and most important parameters are listed below. The whole MATLAB code for the solution for surface evaporation was provided by Dipl.-Ing. Radl. The analytic verification without shrinking is attached in the appendix.

| | |
|---|---|
| Diffusion coefficient | $D_{EA} = 10^{-11} \left[\frac{m^2}{s}\right]$ |
| Mass transfer coefficient | $\beta_{EA} = 2{,}14 \cdot 10^{-4} \left[\frac{m^{-4}}{s}\right]$ |
| Initial volume fraction in the micro-particle | $\varphi_{EA} = 0{,}81$ |
| Equilibrium volume fraction in the liquid phase | $\varphi_{EA,eq} = 0{,}1$ |
| Inner radius | $r_{min} = 10^{-6}$ [m] |
| Outer radius | $r_{max} = 75 \cdot 10^{-6}$ [m] |
| Starting Time | $t_{min} = 0$ [s] |
| Ending Time | $t_{max} = 50$ [s] |
| Number of terms in the summation | $n_{max} = 100$ |
| Number of radial sections | $n_{Radial} = 2500$ |
| Time step | $\Delta t = 1$ [s] |

To compare the concentration used in the equations by Crank [7] and the volume fraction used in our algorithm, the molar density of every component was set to one. This assumption enables a direct comparison between analytical and numerical solution. The equations showed below are all found in the book “Mathematics of Diffusion” by Crank [7].

At the beginning the sphere is initially at a uniform concentration $C_i$ and there is a surface condition:

$$-D_i \frac{\partial C}{\partial r} = \beta_i \cdot (C_S - C_0) \tag{5-33}$$

Where $D_i$ is the binary diffusion coefficient, $\beta_i$ is the mass transport coefficient, r is the radial position, $C_s$ is the actual concentration just within the sphere, and $C_0$ is the concentration required to maintain equilibrium with the surrounding atmosphere. According to Crank [7], the analytical solution for this problem can be written as:

$$\frac{C_S - C_0}{C_i - C_0} = \frac{2 \cdot Bi}{r_n} \cdot \sum_{n=1}^{\infty} \frac{\exp \cdot (-Fo \cdot \alpha_n^2)}{\{\alpha_n^2 + Bi(Bi - 1)\}} \cdot \frac{\sin(\alpha_n \cdot r_n)}{\sin(\alpha_n)} \tag{5-34}$$

Here, $r_n$ is the normalized radius, which is correlated to the maximum radius of the sphere $r_{max}$ as:

$$r_n = \frac{r}{r_{max}} \tag{5-35}$$

The dimensionless mass Fourier number is used in the analysis of unsteady transfer processes [12] and can be expressed as:

$$Fo = \frac{t \cdot D_i}{r_{max}^2} \tag{5-36}$$

The terms $\alpha_n$s are the roots of

$$\alpha_n \cot(\alpha_n) + Bi - 1 = 0 \tag{5-37}$$

Some of the roots can be found in the book “Mathematics of Diffusion” by Crank [7].
The Biot number is defined as:

$$Bi = \frac{r_{max} \cdot \beta_i}{D_i} \tag{5-38}$$

The dimensionless mass transfer Biot number is used to characterize non-steady-state (or transient) mass diffusion processes.

These equations were also solved with MATLAB, and the results were compared with the numerical ones. The comparison of these calculations is presented in Figure 5-2. The deviation between the numeric and analytic results at the inner $r_{min}$ and the outer radius $r_{max}$ is shown in Figure 5-3 and can be obtained from Eqn. (5-39). Here, $\varphi_{Analytical,i,r,t}$ is the analytic and $\varphi_{Numerical,i,r,t}$ is the numerical found value for the volume fraction.

$$Deviation = \left(\frac{\varphi_{Analytical,i,r,t}}{\varphi_{Numerical,i,r,t}} - 1\right) \cdot 100 \tag{5-39}$$

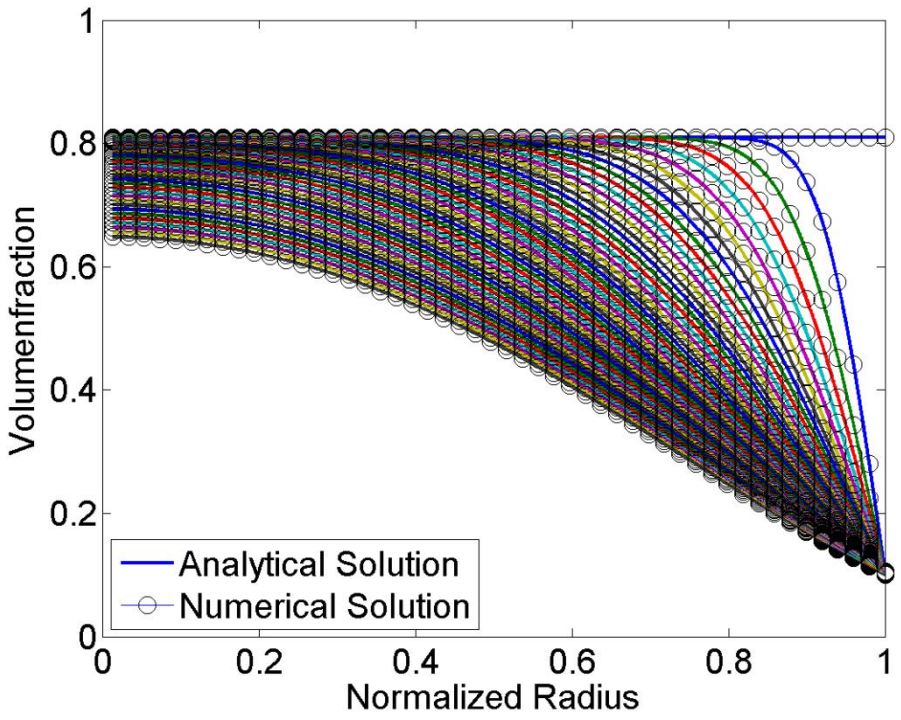



**Figure 5-2: Comparison of analytical solution of Crank [7] and numerical results.**

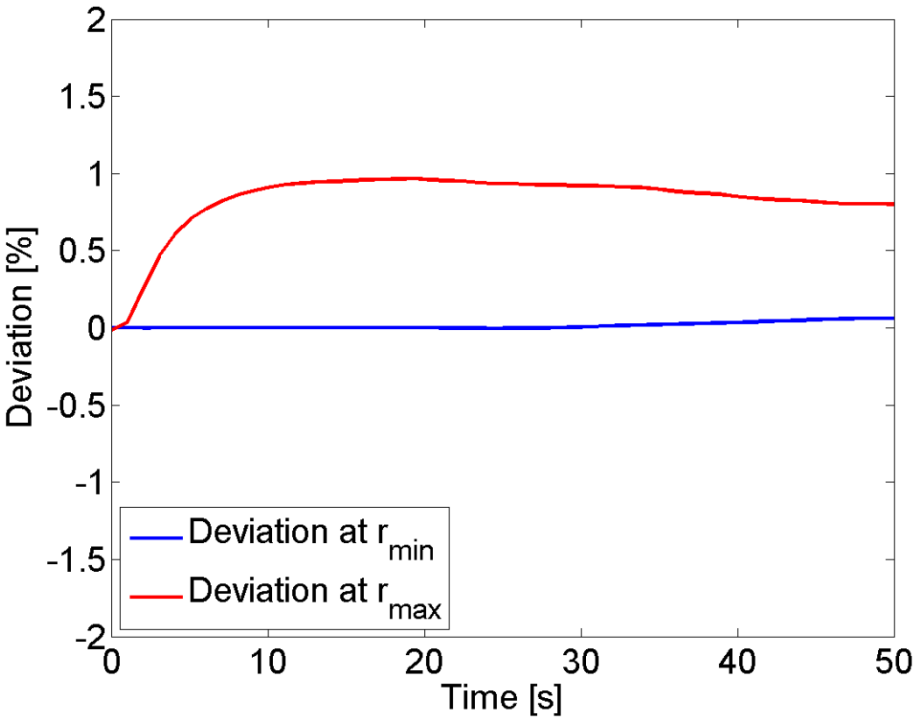



**Figure 5-3: Deviation between the numerical and analytical results at $r_{min}$ and $r_{max}$.**

Figure 5-2 clearly shows that the numerical solution and the analytical algorithm provide nearly the same results. To verify this observation, the deviation between the two algorithms was compared at the inner $r_{min}$ and the outer radius $r_{max}$. In Figure 5-3 the deviation between the analytic and numerical results is shown in percent over time. The occurring deviation of +/- 1 % was assumed to be negligible for the engineering objectives of this work.

## 5.5. Particle Shrinking

For the first calculations, the shrinking of the particles during the hardening process was not taken into account. This assumption is reasonable if the extraction on the surface of the micro-particle is so fast, e.g. under a few seconds, that a hull is formed immediately and, consequently, the diameter of the particles do not significantly change over time. This hull would be then considered the limiting step for the mass transfer.

To prove this assumption a series of experiments were conducted. During the extraction process samples were taken and the micro particle diameter was determined. The two particle size analysis systems used are from Sympatec, namely HELOS, a well proven laser diffraction system, and QICPIC, an image analysis system. In the experimental section of this work a short overview of these two measurement systems is presented.

### 5.5.1. Experimental Results

Figure 5-4 shows experimental results of the micro-particle diameter over time. At the beginning of the experiment (t=0) the diameter of the micro-particle was measured to be equal to 163,49 μm. During the extraction process the diameter of the particle decreased to around 100 μm because the two solvents diffused in the continuous phase. As a consequence of these occurrences, the assumption that the micro-particle size will not change significantly over time has been disproved. Thus, the description of the shrinking process appeared necessary to obtain correct results.

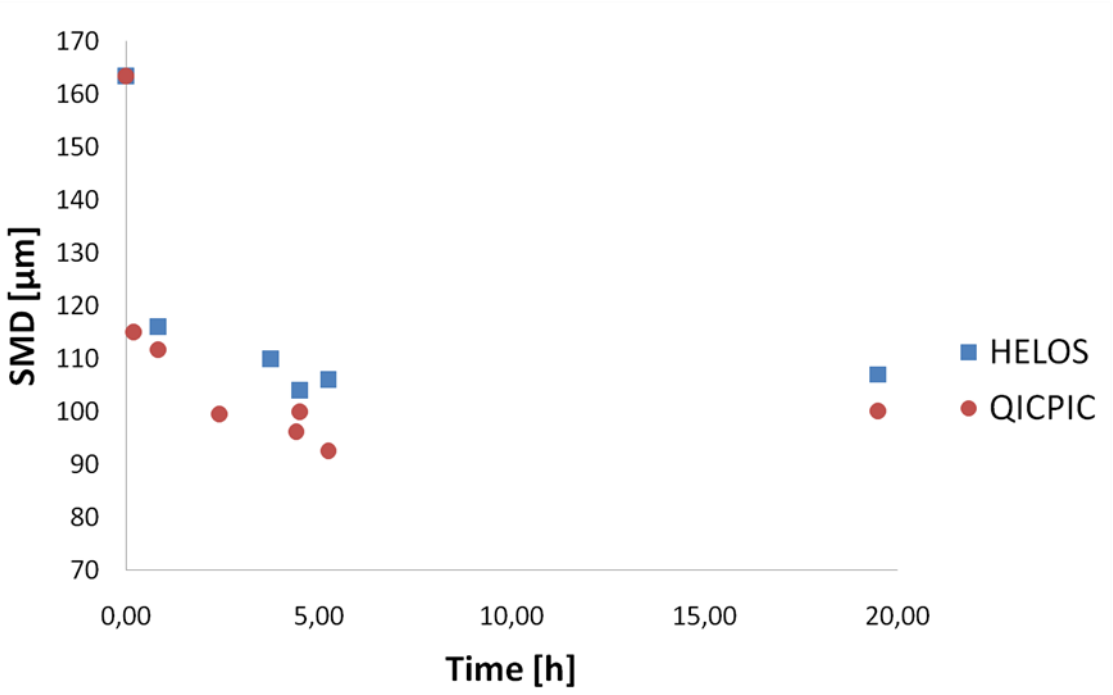



**Figure 5-4: Experimental results for the diameter of the micro-particle over time during the extraction process (Revolutions 260 rpm).**

### 5.5.2. Shrinking Model

As presented above, the shrinking of the particle cannot be considered to be negligible. Therefore, a method to numerically compute the diameter reduction of the micro-particles was developed, which is described in the following sections. The different MATLAB code parts can be found in the appendix.

Basically, the following procedure was used:

1. The differential equation presented in chapter 5.1, for convenience written here again as Eqn. (5-40), was solved for a defined time interval assuming that the shrinking was insignificant during the considered time step.

$$\frac{\partial \varphi_{DP,EA}}{\partial t} = \frac{1}{r^2} \cdot \frac{\partial}{\partial r} \cdot \left( r^2 \cdot D_{DP,EA} \cdot \frac{\partial \varphi_{DP,EA}}{\partial r} \right) \tag{5-40}$$

   The solution of the equation is a three dimensional array which determines the volume fraction $\varphi_i$ of the two solvents at any given radial position $x_n$ respectively $r_n$ and at any time t. As mentioned in section 5.3, the strictly monotonic increasing row vector $x_{mesh}$ defines the x values $x_n$ at which the numerical solution is

computed, while the strictly monotonic increasing row vector $t_{span}$ specifies the time points t where the solution has to be returned.

2. For calculation purposes, the micro-particle was discretized into n different spherical shells. The number of spherical shells is determined by the number of radial sections $n_{Radial}$, which also determines the number of elements of the row vector $x_{mesh}$. This value has an enormous influence on the accuracy of the results and has been investigated in Chapter 6.

   The volume fraction of every spherical shell was determined with the mean volume fractions $\varphi_{i,m,n,t}$ of the two adjacent radial positions $x_{n,t}$ and $x_{n+1,t}$. The equation to calculate the mean volume fraction of a spherical shell at time t is defined as:

$$\varphi_{i,m,n,t} = \frac{\left(\varphi_{i,n,t} + \varphi_{i,n+1,t}\right)}{2} \tag{5-41}$$

   Here, $\varphi_{i,n,t}$ and $\varphi_{i,n+1,t}$ are the volume fractions of a component i, determined by the solution matrix of the differential equation at two specific radial positions $x_{n,t}$ respectively $r_n$ and $x_{n+1,t}$ respectively $r_{n+1,t}$, as well as a certain time t.

   In Figure 5-5 a scheme of a spherical shell at the time t is shown. The continuous line represents one boundary of the exemplary discussed spherical shell with the volume fractions $\varphi_{i,n+1,t}$ at $x_{n+1,t}$ and the radius $r_{n+1,t}$. The dashed line represent the other boundary surface with the volume fractions $\varphi_{i,n,t}$ at $x_{n,t}$ and the radius $r_{n,t}$. $\varphi_{i,m,n,t}$ is the mean volume fraction of the considered spherical shell n+1 obtained from Eqn. (5-41).

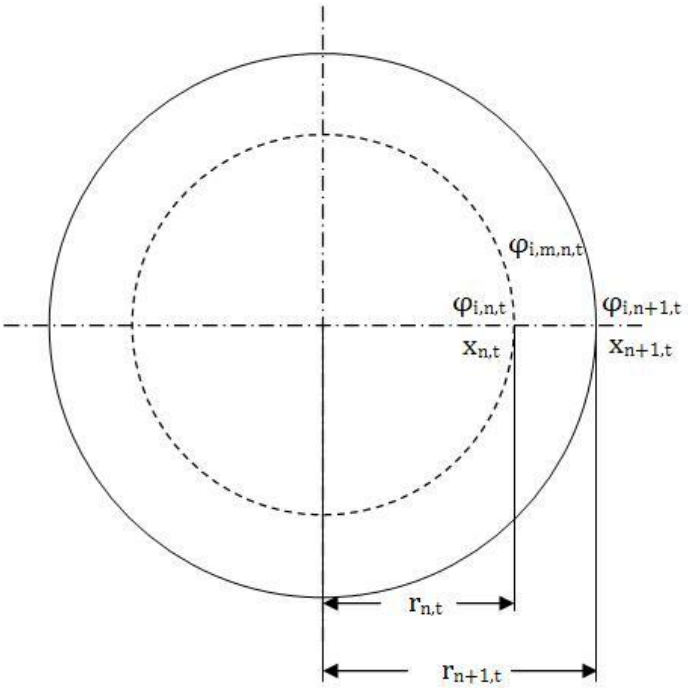



**Figure 5-5: Scheme of a singular spherical shell where the shrinking process is calculated.**

In order to determine the mass loss $m_i$ of a singular particle in a time interval t, the mean volume fractions $\varphi_{i,m,t}$ of the solvents is first transformed into the mass fraction $w_{i,m,t}$ by using Eqn. (5-42) to Eqn. (5-46).

$$\varphi_{i,m,t} = \frac{V_{i,m,t}}{\sum_1^i V_{i,m,t,}} \tag{5-42}$$

$$m_{i,m,t} = \rho_i \cdot V_{i,m,t} \tag{5-43}$$

$$w_{i,m,t} = \frac{m_{i,m,t}}{\sum_1^i m_{i,m,t}} \tag{5-44}$$

$$w_{i,m,t} = \frac{\rho_i \cdot \varphi_{i,m,t} \cdot \sum_1^i V_{i,m,t}}{\sum_1^i m_{i,m,t}} \tag{5-45}$$

$$\frac{\sum_1^i V_{i,m,t}}{\sum_1^i m_{i,m,t}} = \frac{1}{\rho_{shell,m,t}}$$

$$w_i = \frac{\rho_i \cdot \varphi_i}{\rho_{shell,m,t}} \tag{5-46}$$

Here, $\rho_i$ is the density of the component i and $\rho_{Shell,m,t}$ is the density of a spherical shell, $V_{i,m,t}$ is the volume and $m_{i,m,t}$ is the mass of a spherical shell at the time t.

3. Basically, the density of every spherical shell $\rho_{Shell,m,t}$ is not constant and is a function of the volume fractions of the different components. The value of the density thus changes with the extraction of the solvents and is a time-dependent value.

$$\rho_{Shell,m,t} = \varphi_{EA,m,t} \cdot \rho_{EA} + \varphi_{BA,m,t} \cdot \rho_{BA} + \varphi_{PVA,m,t} \cdot \rho_{PVA} + \varphi_{PLGA,m,t} \cdot \rho_{PLGA} + \varphi_{API,m,t} \cdot \rho_{API} \quad (5\text{-}47)$$

The volume of every spherical shell $V_{Shell,m,t=0}$ can be determined by just subtracting the volumes of two neighboring spheres $V_{Sphere,r_{n+1},t=0}$ and $V_{Sphere,r_n,t=0}$ at the time (t=0).

$$V_{Shell,m,t=0} = V_{Sphere,r_{n+1},t=0} - V_{Sphere,r_n,t=0} \quad (5\text{-}48)$$

$$V_{Shell,m,t=0} = \frac{4}{3} \cdot \pi \cdot \left(r^3_{n+1,t=0} - r^3_{n,t=0}\right) \quad (5\text{-}49)$$

The mass of the spherical shell $m_{Shell,m,t=0}$ can be then determined by multiplying the volume of the spherical shell $V_{Shell,m,t=0}$ with its current density $\rho_{Shell,m,t=0}$.

$$m_{Shell,m,t=0} = V_{Shell,m,t=0} \cdot \rho_{Shell,m,t=0} \quad (5\text{-}50)$$

4. The presented steps enable the calculation of the mass difference $\Delta m_{Shell,m,t}$ from one time step to another.

$$\Delta m_{Shell,m,t} = m_{Shell,m,t=0} \cdot \left(w_{i,m,t=0} - w_{i,m,t}\right) \quad (5\text{-}51)$$

Consequently, the new mass $m_{Shell,m,t}$ of every spherical shell can be updated as:

$$m_{Shell,m,t} = m_{Shell,m,t=0} - \Delta m_{Shell,m,t} \quad (5\text{-}52)$$

The volume of each shell $V_{Shell,m,t}$ is then obtained by dividing its mass for its current density.

$$V_{Shell,m,t} = \frac{m_{Shell,m,t}}{\rho_{Shell,m,t}} \tag{5-53}$$

Afterwards, the volume of the new micro-particle $V_{Sphere,t}$ can be obtained from Eqn. (5-54) and the whole volume difference $\Delta V_{Sphere}$ from one time step to another is calculated from Eqn. (5-55).

$$V_{Sphere,t} = \sum_{m=1}^{i} V_{Shell,m,t} \tag{5-54}$$

$$\Delta V_{Sphere} = V_{Sphere,t=0} - V_{Sphere,t} \tag{5-55}$$

5. The new radius of the micro-particle $r_t$ at the time t can now be determined by solving the Eqn. (5-56) to Eqn. (5-58).

$$\Delta V_{Sphere} = V_{Sphere,t=0} - V_{Sphere,t} \tag{5-56}$$

$$\Delta V_{Sphere} = \frac{4}{3} \cdot \pi \cdot r_{t=0}^3 - \frac{4}{3} \cdot \pi \cdot r_t^3 \tag{5-57}$$

$$r_t = \sqrt[3]{r_{t=0}^3 - \sum_{n=1}^{n} \frac{3}{4 \cdot \pi} \cdot \Delta V_{Sphere}} \tag{5-58}$$

Here, $r_{t=0}$ is the radius of the micro-particle at the time (t=0).

6. Finally, the numerical grid is rearranged to the updated radius. This algorithm is solved after each time step of the PDEPE solver from MATLAB. The method seemed not to relevantly affect the computational costs of the simulation.

# 6. Parameter Studies

In this section the effects of different parameters on the extraction/shrinking process are presented in detail. Precisely, the influence of variables like the time step size (Δt), the length of a radial section (Δx) or the system dependent constant ($r_{2i}$ and $r_{3i}$), in the correlation of the diffusion coefficient are presented and discussed in terms of final micro-particle diameter and distribution of the polymer inside the micro-particle.

Concerning the structure of this chapter, the studied parameters are first defined. Afterwards, the influence of the length of a radial section (Δx) and the time step size (Δt) on the results are determined. Then a sensitivity analysis of the correlation of the diffusion coefficient is presented. Finally, a experimental validation of the numerical results was conducted.

## 6.1. Material Properties

The most important material properties of the four different components used in the process are summarized in the following tables. The volume fraction of the different components $\varphi_i$ can be calculated from Eqn. (6-59)

$$\varphi_i = \frac{V_{i,}}{V_{Total}} \tag{6-59}$$

where $V_i$ is the volume of one component and $V_{Total}$ is the total volume of the system which is defined as follows:

$$V_{Total} = \sum_{i=1}^{m} V_i \tag{6-60}$$

The volume fractions at the beginning of the process (t=0) are shown in Table 6-1.

**Table 6-1: Volume fractions of the different components at the beginning of the extraction process.**

| Volume fractions of the different components | | | |
|---|---|---|---|
| $\varphi_{EA,\,t=0}$ | $\varphi_{BA,\,t=0}$ | $\varphi_{Polymer,\,t=0}$ | $\varphi_{API,\,t=0}$ |
| **[$m^3$ EA/$m^3$ DP]** | **[$m^3$ BA/$m^3$ DP]** | **[$m^3$ Polymer/$m^3$ DP]** | **[$m^3$ API/$m^3$ DP]** |
| 0,618 | 0,2294 | 0,1114 | 0,0412 |

The mass fraction of the different components $w_i$ can be calculated from Eqn. (6-61).

$$w_i = \frac{\varphi_i \cdot \rho_{i,}}{\rho_{DP}} \tag{6-61}$$

Here, $\rho_i$ is the density of one component and $\rho_{DP}$ is the density of the disperse phase.
The mass fractions at the beginning (t=0) are shown in Table 6-2.

**Table 6-2: Mass fractions of the different components at the beginning of the extraction process.**

| Mass fractions of the different components | | | |
|---|---|---|---|
| $w_{EA,\,t=0}$ | $w_{BA,\,t=0}$ | $w_{Polymer,\,t=0}$ | $w_{API,\,t=0}$ |
| **[kg EA/kg DP]** | **[kg BA/kg DP]** | **[kg Polymer/kg DP]** | **[kg API/kg DP]** |
| 0,5695 | 0,2405 | 0,1120 | 0,0780 |

All the material parameters needed for the simulations refer to Perry's Chemical Engineers' Handbook [26], the VDI Wärmeatlas [34] or the sticker on the chemicals and are summarized in Table 6-3.

**Table 6-3: Material Parameters.**

| | | | Ethyl acetate | Benzyl alcohol | Water |
|---|---|---|---|---|---|
| **Molecular weight** | **M** | **[g/mol]** | 88,10 | 108,14 | 18,02 |
| **Melting point** | **$T_m$** | **[°C]** | -82,4 | -15,3 | 0,01 |
| **Boiling Point** | **$T_B$** | **[°C]** | 77,1 | 204,7 | 99,97 |
| **Density** | **ρ** | **[g/$cm^3$]** | 0,901 | 1,046 | 1,00 |
| **Dynamic Viscosity** | **η** | **[mPa s]** | -- | -- | 1,518 |

The densities and viscosities of disperse and continuous phase are shown in Table 6-4 and were determined experimentally using a shear rheometer (Physica MCR 301) and a digital densitymeter (DMA 38) from the Anton Paar GmbH.

**Table 6-4: Densities and viscosities of the disperse and continuous phase.**

| | **Density** **ρ** **[g/cm3]** | **Dynamic Viscosity** **η** **[mPa s]** |
|---|---|---|
| **Disperse Phase** | 0.9948 | 174,01 |
| **Continuous Phase** | 0.9993 | 1,419 |

## 6.2. Sensitivity Study for the Time Step Size and the Number of Radial Sections

As described beforehand the differential equation was solved for a determined time interval assuming that the shrinking was insignificant during the considered time step. Hence, the influences of the time step size and the length of one radial section on the accuracy of the results were first investigated.

The length of one radial section (Δx) can be calculated from equation (4-4).

$$\Delta x = \frac{d_t}{n_{Radial}} \tag{6-62}$$

Here, $d_t$ is the diameter of the micro-particle at the time (t) and $n_{Radial}$ is the number of radial sections.

The most important simulation parameters for the sensitivity study are shown in Table 6-5.

**Table 6-5: Simulation parameters for the sensitivity study.**

| Diameter | Time | Revolution rate | Diffusion coefficient | |
|---|---|---|---|---|
| $d_{t=0}$ | $t_{total}$ | n | $D_{EA,t=0}$ | $D_{BA,t=0}$ |
| [µm] | [s] | [rpm] | [$m^2/s$] | [$m^2/s$] |
| 163,49 | 50 | 260 | $1{,}750 \cdot 10^{-7}$ | $1{,}692 \cdot 10^{-7}$ |

The values of the system-dependent constants $r_1$, $r_2$ and $r_3$, which are needed to calculate the concentration-depending diffusion coefficient, are assumed to be equal to the values used by Reuvers et al. [25], (see Table 6-6). A detailed analysis of the influence of these coefficients is presented in section 1.4.

**Table 6-6: System-dependent constants for the calculation of the diffusion coefficient.**

| System-dependent constants | | | | | |
|---|---|---|---|---|---|
| $r_{EA,1}$ | $r_{EA,2}$ | $r_{EA,3}$ | $r_{BA,1}$ | $r_{BA,2}$ | $r_{BA,3}$ |
| 1,750 | 5,75 | 11.25 | 1,692 | 5,75 | 11,25 |

The investigated ranges of the various parameters are shown in Table 6-7 and Table 6-8.

**Table 6-7: Investigated range of time step sizes (Δt).**

| Time step<br>Δt<br>[sec] |
|---|
| 0,05 |
| 0,1 |
| 0,5 |
| 1 |

**Table 6-8: Investigated range of radial sections lengths (Δx).**

| Number of Radial sections<br>$n_{Radial}$ | Length of a radial section<br>Δx<br>[µm] |
|---|---|
| 20 | 8,175 |
| 50 | 3,269 |
| 100 | 1,635 |
| 200 | 0,817 |
| 400 | 0,409 |

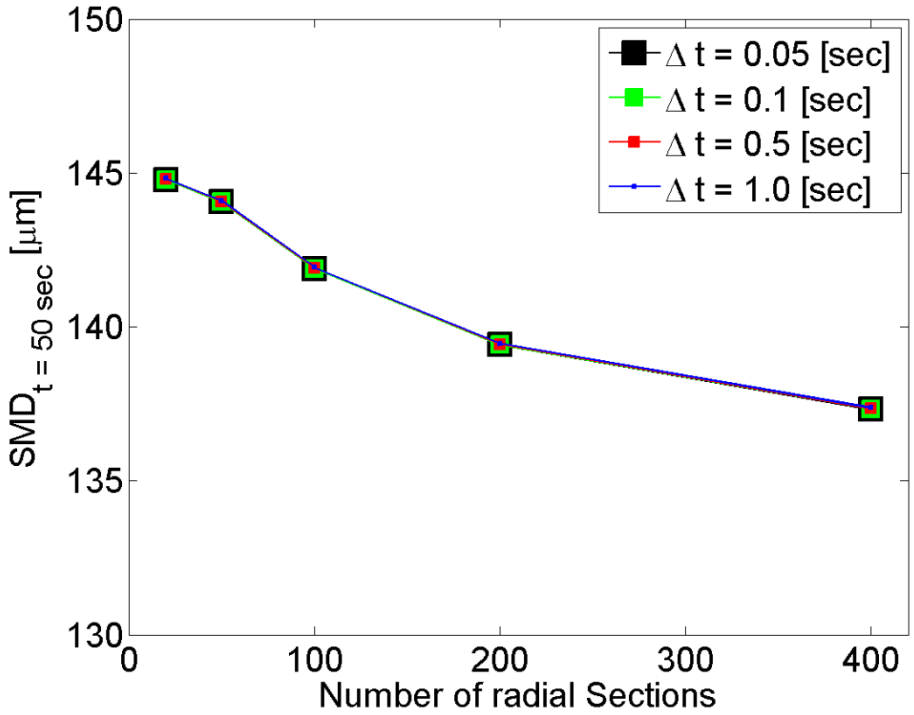



**Figure 6-1: Dependency of the SMD on the number of radial sections ($n_{Radial}$).**

The influence of the number of radial sections ($n_{Radial}$) on the SMD after 50 s of extraction is illustrated in Figure 6-1. With the increase of the number of radial sections the SMD decreases. As also shown in Figure 6-1, the influence of the time step size (Δt) on the SMD appears insignificant in the investigated range. Notice, that if the number of radial sections is doubled from 200 to 400 the results change by only one percent. The calculation time instead has to be increased by a factor of 5. Thus, the enhancement due to finer grid was considered to be insignificant for the engineering objectives of this work. As a consequence, all further simulations were conducted with 200 radial sections, which correspond to a length of a singular section of 0,817 µm.

In Figure 6-2 the SMD after 50 s of extraction over the different time step sizes (Δt) is shown. This figure clearly shows that the influence of the time step size on the final micro-particle diameter is insignificant in the investigated range.

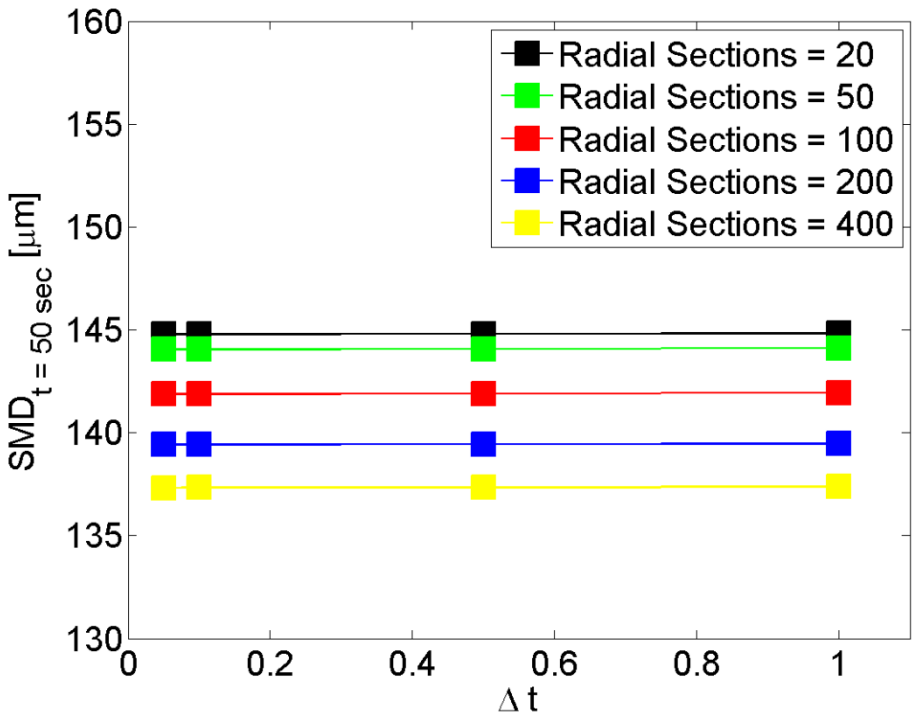



**Figure 6-2: Dependency of the SMD on the time step size (Δt).**

## 6.3. CFL-Analysis

In this section the combined effect of the time step size and the grid refinement on the accuracy of the results was further investigated.
The CFL-number or Courant-number is used in the numerical flow simulation for the discretization of time dependent partial differential equations. This parameter is named after Richard Courant, Kurt Friedrichs, and Hans Lewy [6]. It arises when explicit time-marching schemes are used for the numerical solution. To avoid incorrect results, the time step must be less than a certain time in many explicit time-marching computer simulations.

In our case we simply use this parameter to connect the sizes of time step and radial sections with a well-known parameter. Nevertheless, it has not to be confused with the stability analysis for the CFD simulations.

The value of the CFL-number (6-63) is written as follows:

$$\mathrm{CFL} < \frac{v \cdot \Delta t}{\Delta x} \tag{6-63}$$

where v is the velocity at the beginning of the extraction process, which can be obtained from Eqn. (6-64).

$$v = \frac{\dot{M}_{t=0}}{A_P \cdot \rho_{DP}} \tag{6-64}$$

Here, $\dot{M}_{t=0}$ is the mass transfer rate at (t=0), $A_P$ is the surface of the micro-particle and $\rho_{DP}$ is the density of the disperse phase.

The mass transfer rate at the beginning can be defined as:

$$\dot{M}_{t=0} = \beta_i \cdot A_p \cdot \Delta c \tag{6-65}$$

where $\beta_i$ is the mass transport coefficient of the different diffusing solvents and $\Delta c$ is the concentration difference between the continuous and disperse phase.
The concentrations of the different solvents at the interface can be expressed trough their volume fractions $\varphi_i$ and their densities $\rho_i$ and the concentration in the continuous phase at

the beginning can be assumed to be equal to zero. Finally, the mass transfer rate is calculated by using Eqn. (6-66) to Eqn. (6-67):

$$\dot{M}_{t=0} = \dot{M}_{EA,t=0} + \dot{M}_{BA,t=0}$$

$$\dot{M}_{t=0} = \beta_{EA} \cdot A_P \cdot \left(\varphi^*_{EA} \cdot \rho_{EA} + c_{CP,EA}\right) + \beta_{BA} \cdot A_P \cdot \left(\varphi^*_{BA} \cdot \rho_{BA} + c_{CP,BA}\right) \tag{6-66}$$

$$v = \frac{\beta_{EA} \cdot (\varphi^*_{EA} \cdot \rho_{EA}) + \beta_{BA} \cdot (\varphi^*_{BA} \cdot \rho_{BA})}{\rho_{DP}} \tag{6-67}$$

Here, $\varphi_i^*$ is the volume fraction of the component i at the interface.

The calculated values for the velocity and the corresponding CFL-numbers are shown in Table 6-9.

**Table 6-9: CFL-numbers for the different calculated cases.**

| CFL-Number | Velocity v [m/s] | Length of a radial section Δx [µm] | Time step size Δt [s] |
|---|---|---|---|
| 3,24 | 0,0053 | $\frac{D}{200} = 0{,}817$ | 0,0005 |
| 6,48 | | | 0,001 |
| 647,74 | | | 0,1 |
| 6477,45 | | | 1 |

To ensure the comparability of the two analyses, the same simulation parameters and system-dependent constants used in the previous paragraph were used for the CFL-analysis. Hence, the most important simulation parameters are shown in Table 6-5 and the values of the system-dependent constants are presented in Table 6-6.

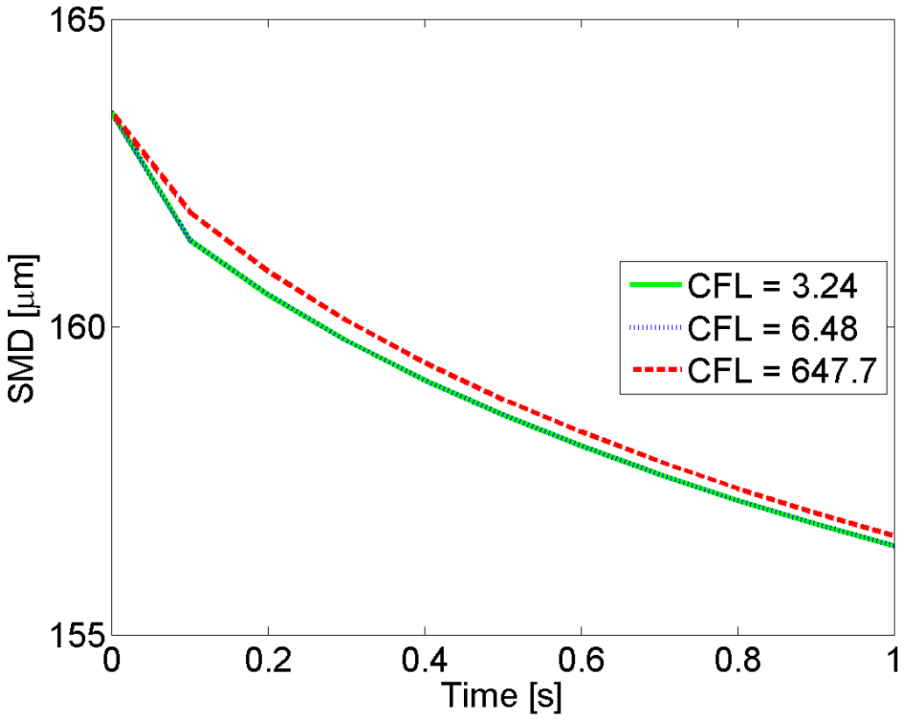



**Figure 6-3: SMD over time for different CFL-numbers.**

In Figure 6-3 the time evolution of the Sauter mean diameter (SMD) for different CFL-numbers is shown. According to these results, it appears evident that the decrease of the CFL-number has no significant influence on the results. To verify this observation, the deviation between the different calculated diameters after one second of extraction process was compared. In Figure 6-4 the deviation in percent is shown over the different CFL-numbers. The occurring deviation of about +/- 1,2% was assumed to be negligible for the engineering objectives of this work.

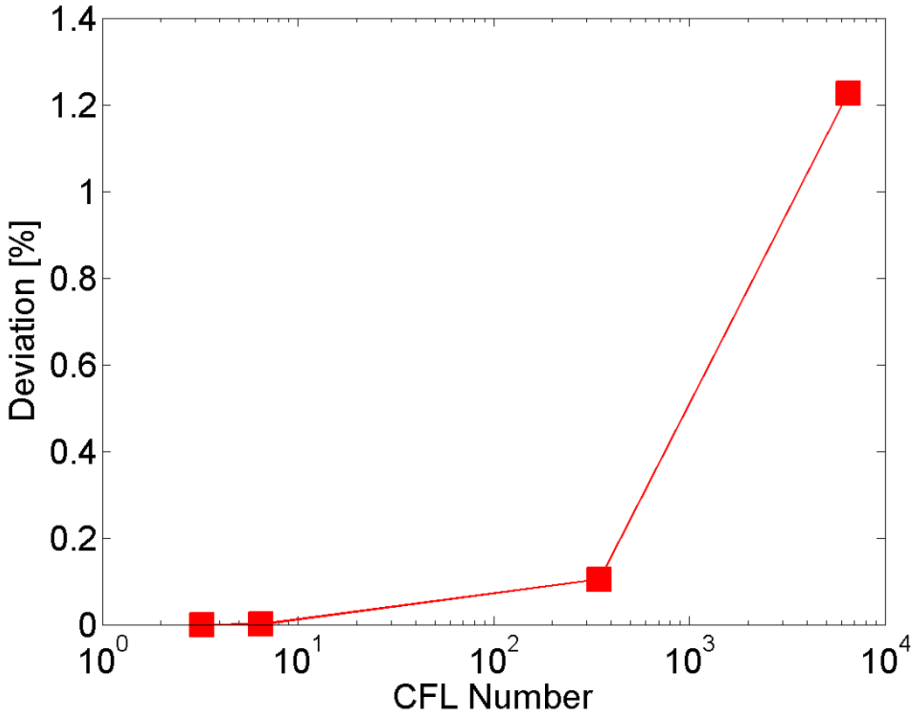



**Figure 6-4: Deviation between the different diameters after one second of the extraction process.**

The influence of the CFL-number on the calculation time is illustrated in Figure 6-5. As shown in the figure, the computational effort for the simulation of one second of the extraction process increased from about 40 seconds at a CFL-number of 6477,4 to about 35 minutes at a CFL-number of 3,24.

As a consequence of the presented facts, all further simulations were conducted with a time step size of one second.

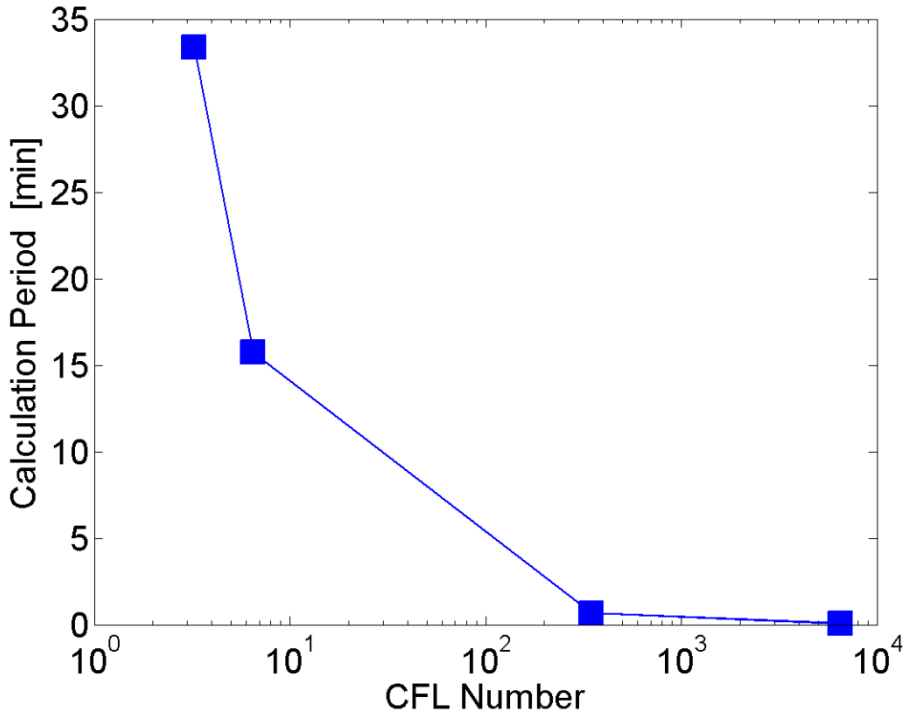



**Figure 6-5: Calculation period for one second of the extraction process for different CFL-numbers.**

In Table 6-10 the most important results of the CFL-analysis are summarized again.

**Table 6-10: Results of the CFL-analysis.**

| **CFL-number** | **$SMD_{after\ 1\ second}$** | **Calculation period** |
|---|---|---|
| **CFL** | **$d_{t=1}$** | |
| | **[µm]** | **[min]** |
| 3,24 | 1,5645 | 33,37 |
| 6,48 | 1,5645 | 15,73 |
| 647,74 | 1,5661 | 0,66 |
| 6477,45 | 1,5836 | 0,07 |

## 6.4. Sensitivity Analysis for the Diffusion Coefficient

As also mentioned in section 5.2.4, the diffusion coefficient for both solvents is highly depending on the polymer volume fraction in the micro-particle. The correlation from Reuvers et al. [25] in a paper by Li et al. [18] for convenience written here again as Eqn (6-68):

$$D_i = r_{1,i} \cdot 10^{-(r_{2,i} + r_{3,i} \cdot \varphi_P)} \tag{6-68}$$

where $r_{1,i}$, $r_{2,i}$ and $r_{3,i}$ are the system-dependent constants and $\varphi_P$ is the volume fraction of the polymer.

The rule of thumb of specifying the $r_i$ values is to adjust these constant to result in a reasonable diffusion coeffcient [18]. Thus, the influences of these system-dependent constants on the results are investigated in this section.

The two system-dependent constants $r_{1,i}$ and $r_{2,i}$ were determined by equating the correlation for the diffusion coefficient from Wilke and Chang [36] with the one from Reuvers et al. [25] at the beginning of the extraction process (t=0). The correlation from Wilke and Chang [36], which is valid for diffusion of various substances in water and in non-associated solvents, is for convenience written here again as Eqn. (6-69).

$$D_i = 7{,}4 \cdot 10^{-8} \cdot \frac{(x \cdot M)^{\frac{1}{2}} \cdot T}{\eta \cdot V^{0,6}} \tag{6-69}$$

Here, M is the molecular weight of the solvent, T the temperature, $\eta$ is the viscosity of solution and V is the molar volume of solute at normal boiling point.

The obtained values for system-dependent constants $r_{1,i}$ and $r_{2,i}$ for the two solvents are shown in Table 6-11.

**Table 6-11: System-dependent constants for the correlation of diffusion coefficient for the sensitivity analysis.**

| Ethyl acetate | | Benzyl alcohol | |
|---|---|---|---|
| System-dependent constants | | | |
| $r_{1,EA}$ | $r_{2,EA}$ | $r_{1,BA}$ | $r_{2,BA}$ |
| 1,750 | 7,0 | 1,692 | 7,0 |

For a first anlysis, the value of $r_{3,EA}$ for ethyl acetate was set to be equal to the value of $r_{3,BA}$ for benzyl alcohol. The corresponding diffusion coeffiecients of the two solvents at the beginning ($\varphi_{P,t=0}$) and at a theoretical maximal concentration of polymer ($\varphi_P$=1) are shown in Table 6-12.

**Table 6-12: System-dependent constants and corresponding diffusion coefficients for the two solvents.**

| System-dependent constants | | Diffusion coefficients | | | |
|---|---|---|---|---|---|
| $r_{3,EA}$ | $r_{3,BA}$ | $D_{EA,\varphi P,t=0}$ [$m^2/s$] | $D_{EA,\varphi P=1}$ [$m^2/s$] | $D_{BA,\varphi P,t=0}$ [$m^2/s$] | $D_{BA,\varphi P=1}$ [$m^2/s$] |
| **0** | | $1{,}75 \cdot 10^{-7}$ | $1{,}75 \cdot 10^{-7}$ | $1{,}69 \cdot 10^{-7}$ | $1{,}69 \cdot 10^{-7}$ |
| **2** | | $1{,}05 \cdot 10^{-7}$ | $1{,}75 \cdot 10^{-9}$ | $1{,}01 \cdot 10^{-7}$ | $1{,}69 \cdot 10^{-9}$ |
| **4** | | $6{,}27 \cdot 10^{-8}$ | $1{,}75 \cdot 10^{-11}$ | $6{,}06 \cdot 10^{-8}$ | $1{,}69 \cdot 10^{-11}$ |
| **6** | | $3{,}76 \cdot 10^{-8}$ | $1{,}75 \cdot 10^{-13}$ | $3{,}63 \cdot 10^{-8}$ | $1{,}69 \cdot 10^{-13}$ |
| **8** | | $2{,}25 \cdot 10^{-8}$ | $1{,}75 \cdot 10^{-15}$ | $2{,}17 \cdot 10^{-8}$ | $1{,}69 \cdot 10^{-15}$ |
| **10** | | $1{,}35 \cdot 10^{-8}$ | $1{,}75 \cdot 10^{-17}$ | $1{,}30 \cdot 10^{-8}$ | $1{,}69 \cdot 10^{-17}$ |
| **12** | | $8{,}06 \cdot 10^{-9}$ | $1{,}75 \cdot 10^{-19}$ | $7{,}79 \cdot 10^{-9}$ | $1{,}69 \cdot 10^{-19}$ |

The most important simulation parameters for the sensitivity analyses are shown in Table 6-13. These data come again from experimental analysis of the process and are the so called standard parameters.

**Table 6-13: Simulation parameters for the sensitivity analysis.**

| Diameter $d_{t=0}$ [µm] | Time $t_{total}$ [s] | Revolution rate n [rpm] | Time step Δt [sec] | Length of a radial section Δx [µm] |
|---|---|---|---|---|
| 163,49 | 10 | 260 | 1 | 0,817 |

In Figure 6-6 the influences of the values of the system-dependent constants $r_{3,EA}$ and $r_{3,BA}$ on the SMD are shown. It appears evident that the values of these constants have an enourmous influence on the final diameter of the micro-particle. In fact, if the constants $r_{3,EA}$ and $r_{3,BA}$ are for example set to be equal to zero, represented by the black colored line, the diffusion coefficients remain constant to a very high value. As a consequence of that the solvents diffuse immediatly in the continuous phase and the diameter of the micro-particle decreases fastly to its final value.

On the other hand, if the constants $r_{3,EA}$ and $r_{3,BA}$ are set to be equal to twelve, represented by the magenta colored line, the diffusion coefficients change with the variation of the polymer volume fraction inside the micro-particle. As a consequence of that, the diameter of the micro-particle does not decrease so quickly and the extraction is limited by the mass transport rate to the interface inside the micro-particle.

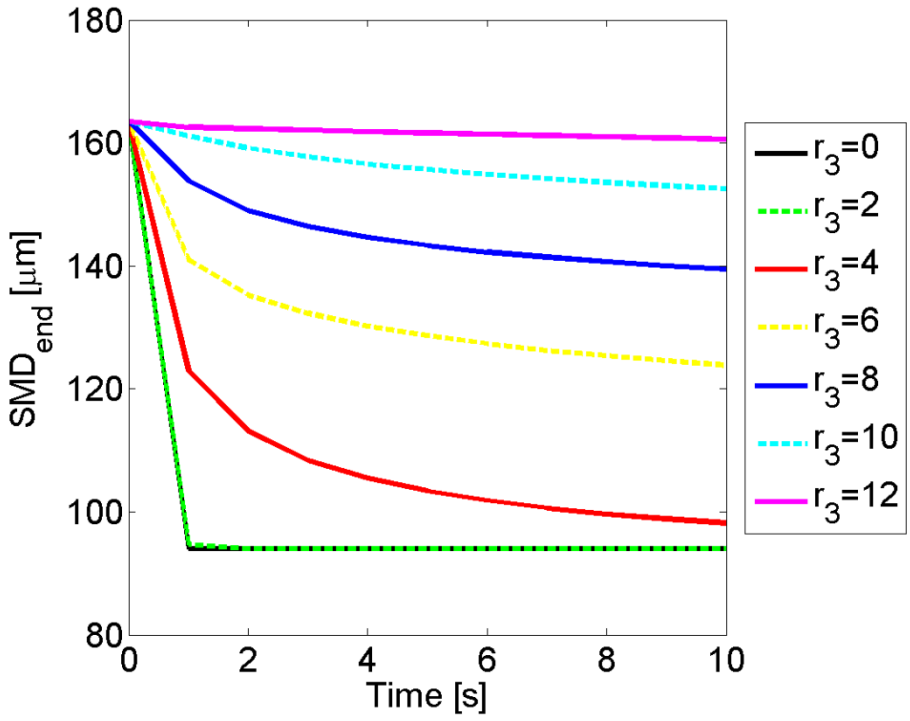



**Figure 6-6: Particle SMD over time for different values of $r_3$.**

In Figure 6-7 the volume fraction of the remaining components (API, Polymer) after a extraction time of ten seconds is shown over the normalized radius. As mentioned above, the solvents diffuse immediatley when the constants $r_{3,EA}$ and $r_{3,BA}$ are set to be equal to zero. This is also evident by the black colored line in Figure 6-7, where the volume fraction of the remaining components increases immediatley and converges to one.

A complete different behaviour is observed when the constants $r_{3,EA}$ and $r_{3,BA}$ are set to be equal to twelve. In this case the diffusion coefficient decreases to a very low value with the increase of the volume fraction of polymer. Hence the hardening process respectivley the extraction of the solvents is limited by the mass transport, inside the micro-particle to the interface. As shown in the figure and represented by the magenta colored line, the volume fraction inside the micro-particles remain almost constant after 10 seconds of extraction. Only at the surface the two solvents are extracted immediatley and the diffusion coefficients decrease to a small values.

This can suggest the formation of a shell and indicates that the first seconds of the extraction process determine the final characteristics of the micro-particle. This assumption is reasonable because this behavior was also observed experimentally. It was supposed that the first seconds that means the first contact from the emulsion droplets with the continuous phase have a great influence on the final characteristics of the micro-particles.

However, further experimentally analysis is required to prove these assumptions.

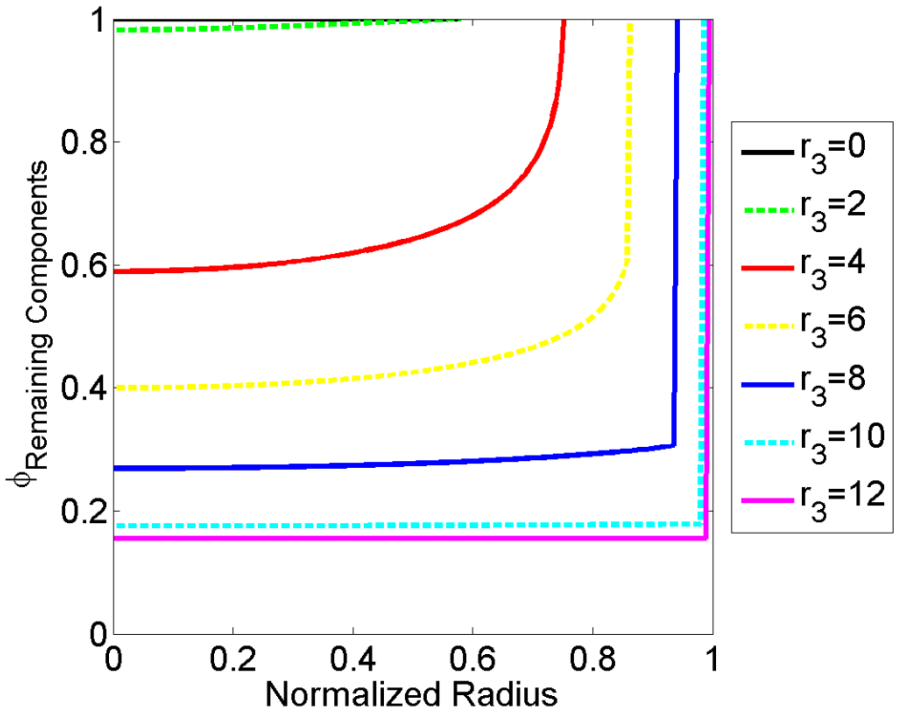



**Figure 6-7: Volume fraction of the remaining components $\varphi_{RP}$ over the normalized radius for different values of $r_3$ after a extraction time of 10 s.**

## 6.5. Experimental Validation

In the previous sections a numerical model has been presented, which is able to simulate the diffusion process of two solvents during the hardening/extraction step, as well as the consequent shrinking of the particles in a stirred reactor. Now, the validation of the method is performed by comparing the numerical results with the data obtained by different experiments performed in laboratory scale.

The prediction of the final micro-particle size was considered to be one of the most important quality attribute. Therefore, the Sauter mean diameters of the micro-particles were determined at different instants of time by two different particle size analyzers, namely HELOS and QICPIC from the Sympatec GmbH. A detailed description of the experimental work and of these two measurement systems can be found in Chapter 6. The experimental set up is shown and described in Chapter 3.

For comparability purposes, the main request was that all the important parameters in the simulation, namely revolution rate of the stirrer, diameter of the micro-particle at the beginning (t=0), as well as material properties, had to be the same as the one used in the experiment. In Table 6-14 the most important simulation parameters for the experimental validation are shown.

**Table 6-14: Simulation parameters for the experimental validation.**

| **Diameter** $d_{t=0}$ [µm] | **Time** $t_{total}$ [s] | **Revolution rate** n [rpm] | **Time step** Δt [sec] | **Length of radial sections** Δx [µm] |
|---|---|---|---|---|
| 163,49 | 50 | 260 | 1 | 0,817 |

As discussed in section 5.2.4, the rule of thumb is to fit the system-dependent constants ($r_{1,i}$ $r_{2,i}$ and $r_{3,i}$) to the experimental data. These values were determined on the one hand by equating the correlation for the diffusion coefficient from Wilke and Chang [36] with the one from Reuver et al. [25] and on the other hand by setting reasonable final values for the diffusion coefficient in the hardened micro-particle. These values and their corresponding diffusion coefficients at the beginning ($\varphi_{P,t=0}$) and at a theoretical maximal concentration of polymer ($\varphi_P$=1) are shown in Table 6-15 and Table 6-16.

**Table 6-15: System-dependent constants and corresponding diffusion coefficients for ethyl acetate.**

| System-dependent constants | | | Diffusion coefficients | |
|---|---|---|---|---|
| $r_{1,EA}$ | $r_{2,EA}$ | $r_{3,EA}$ | $D_{EA,\varphi_{P,t=0}}$ [$m^2/s$] | $D_{EA,\varphi_{P=1}}$ [$m^2/s$] |
| 1,75 | 6,25 | 6,75 | $1{,}75 \cdot 10^{-7}$ | $1.75 \cdot 10^{-13}$ |
| 1,75 | 6,00 | 9,00 | $1{,}75 \cdot 10^{-7}$ | $1{,}75 \cdot 10^{-15}$ |
| 1,75 | 5,75 | 11,25 | $1{,}75 \cdot 10^{-7}$ | $1{,}75 \cdot 10^{-17}$ |

**Table 6-16: System-dependent constants and corresponding diffusion coefficients for the benzyl alcohol.**

| System-dependent constants | | | Diffusion coefficients | |
|---|---|---|---|---|
| $r_{1,BA}$ | $r_{2,BA}$ | $r_{3,BA}$ | $D_{BA,\varphi_{P,t=0}}$ [$m^2/s$] | $D_{BA,\varphi_{P=1}}$ [$m^2/s$] |
| 1,70 | 6,25 | 6,75 | $1{,}69 \cdot 10^{-7}$ | $1{,}70 \cdot 10^{-13}$ |
| 1,70 | 6,00 | 9,00 | $1{,}69 \cdot 10^{-7}$ | $1{,}70 \cdot 10^{-15}$ |
| 1,70 | 5,75 | 11,25 | $1{,}69 \cdot 10^{-7}$ | $1{,}70 \cdot 10^{-17}$ |

The comparison between the numeric and experimental results are illustrated for different extraction times in Figure 6-8 (1 hour of extraction), Figure 6-10 (5 hours of extraction) and Figure 6-9 (20 hours of extraction). In these figures the Sauter mean diameter (SMD) is shown over the time. The colored markers represent the experimental data and the different colored lines the numeric results.

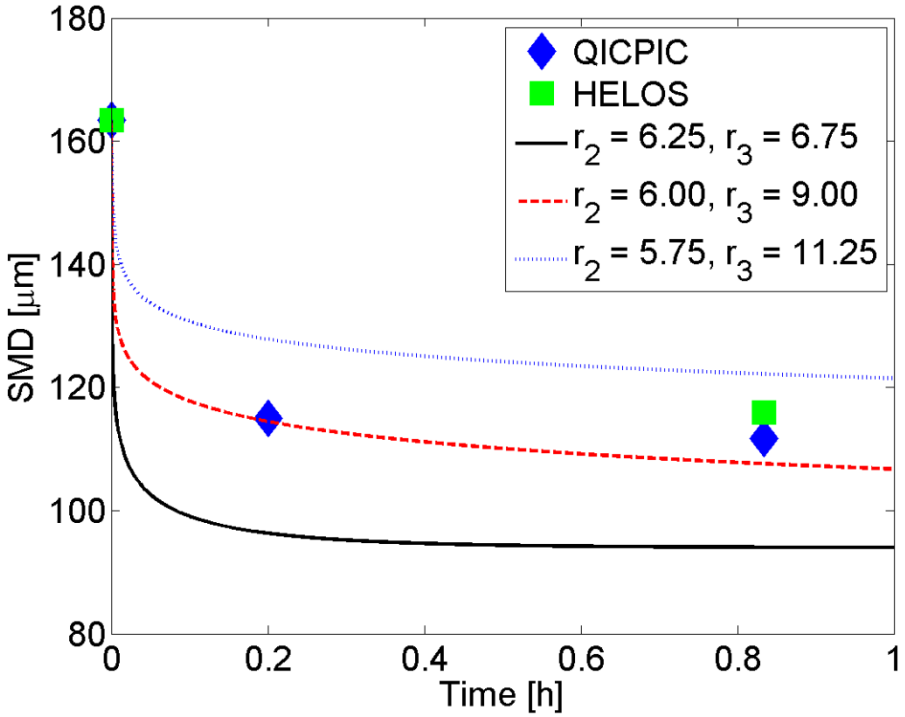



**Figure 6-8: Comparison between the numerical and experimental data over an extraction time of 1 hour.**

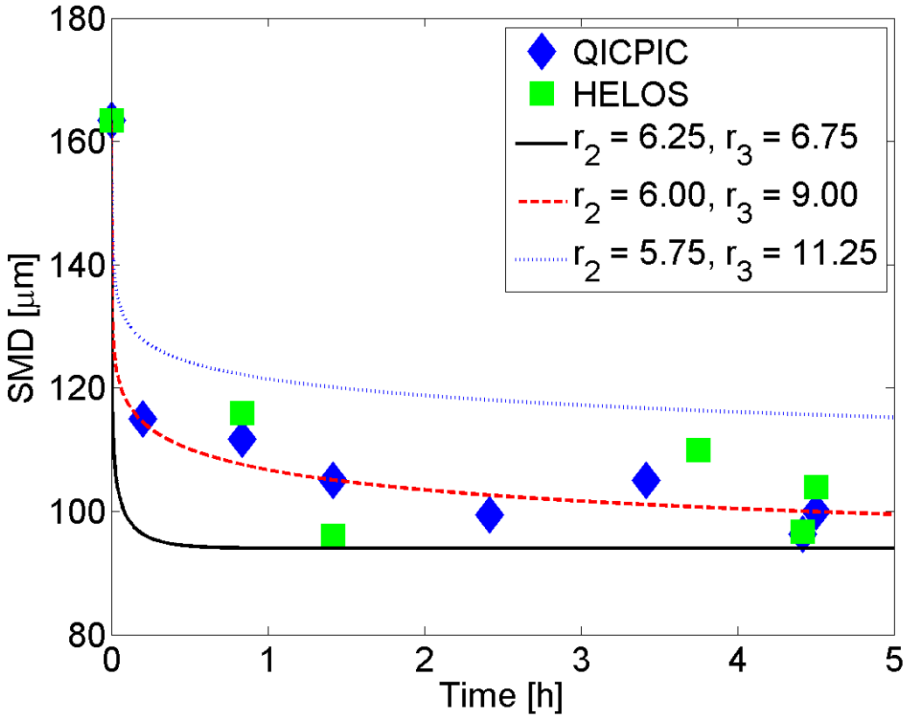



**Figure 6-9: Comparison between the numerical and experimental data over an extraction time of 5 hours.**

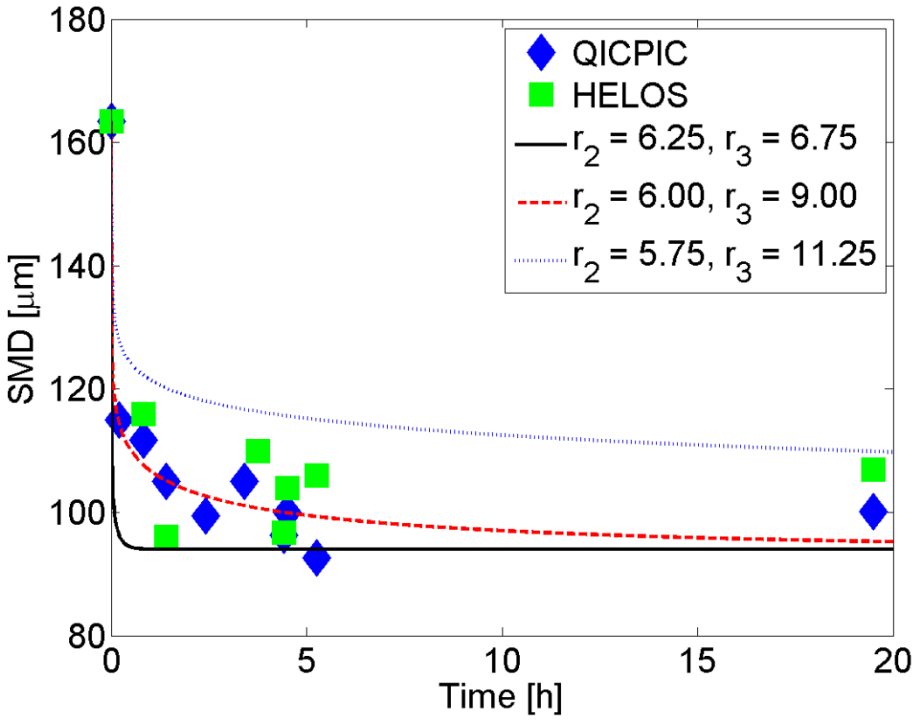



**Figure 6-10: Comparison between the numerical and experimental data over an extraction time of 20 hours.**

As shown in the three figures it is evident that the best match between numerical and experimental data is achieved when the system-dependent constants $r_{3,EA}$ and $r_{3,BA}$ of the two solvents are set to be equal to nine and the $r_{2,EA}$ and $r_{2,BA}$ are set to be equal to six. Thus, we used these values for further analysis of the hardening/extraction step for the material system we considered.

The influences of the system-dependent constants ($r_{2,EA}$, $r_{2,BA}$, $r_{3,EA}$ and $r_{3,BA}$) on the distribution of the volume fractions of the two solvents (ethyl acetate and benzyl alcohol) and the remaining components (API, Polymer) are shown in Figure 6-11, Figure 6-12 and Figure 6-13. The volume fractions are shown for four different instants of time:

- The black colored lines represent the distribution of the different volume fractions at the beginning (t = 0 hour).
- The blue colored lines show the distribution of the different volume fractions after one hour of extraction.
- The red colored lines show the distribution of the different volume fractions after five hours of extraction.

- The magenta colored lines show the distribution of the different volume fractions after twenty hours of extraction.

These three figures clearly show the same behavior as observed by the sensitivity analyses presented in subsection 5.4. For readability reason only the system-dependent constant $r_{3,i}$ is shown in the legend of these figures. The corresponding values of $r_{2,i}$ can be found in Table 6-15 and Table 6-16.

With the increase of the value of the system-dependent constants, the mass transport inside the micro-particle to the interface becomes more and more the limiting step. Hence, the solvents cannot diffuse that quickly and remain inside. In this case the volume fraction is represented by the line with the upward-pointing triangle. In short the volume fractions of the solvents only converge at the interface against zero.

A complete different behavior is observed for the other two cases. Here the solvents diffuse very quickly out of the micro-particle in the continuous phase. As a consequence of that the residual solvent content is very low after 20 hours of extraction.

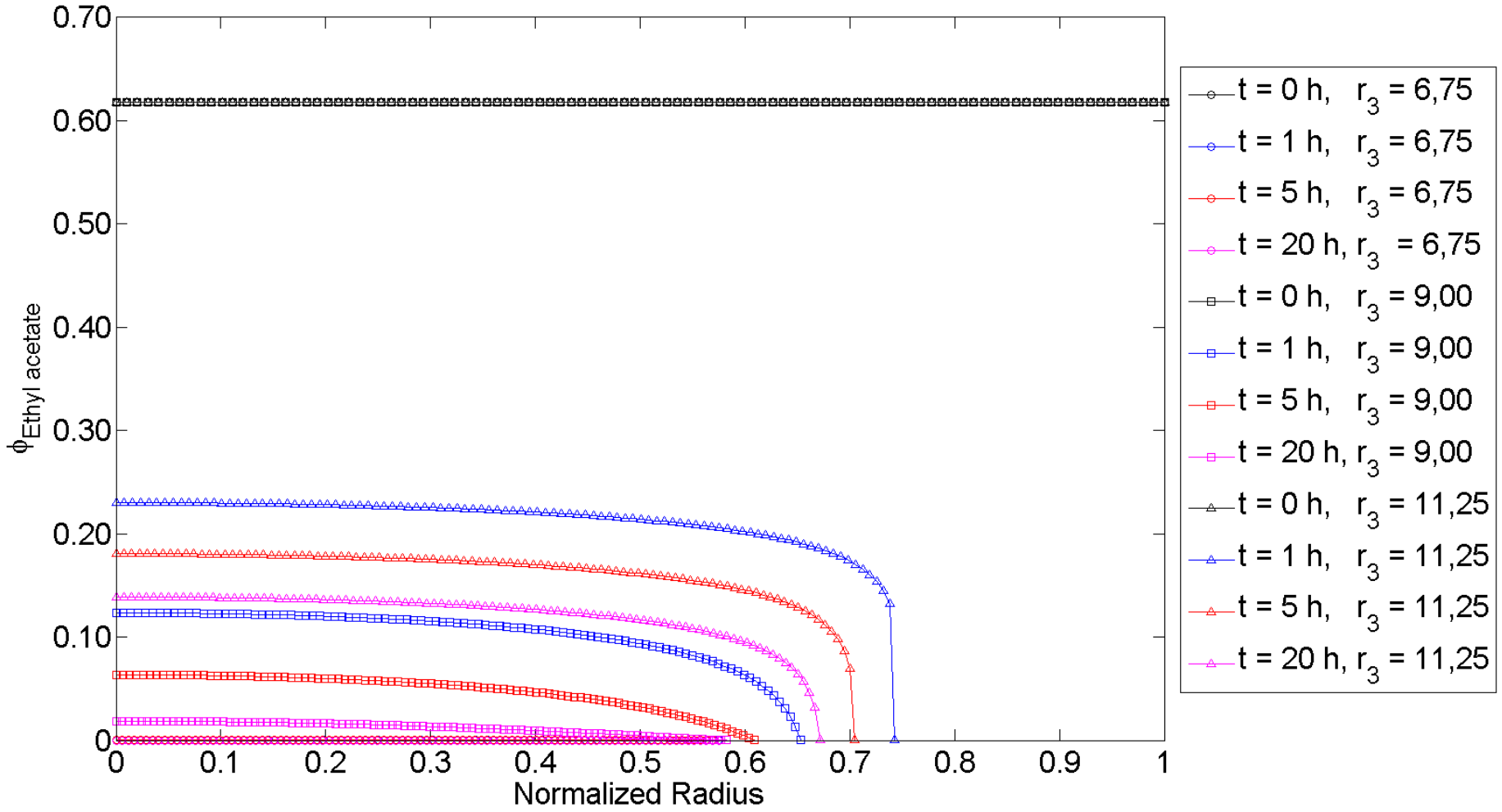



**Figure 6-11: Volume fraction of ethyl acetate over the normalized radius for different values of $r_3$ at different instants of time.**

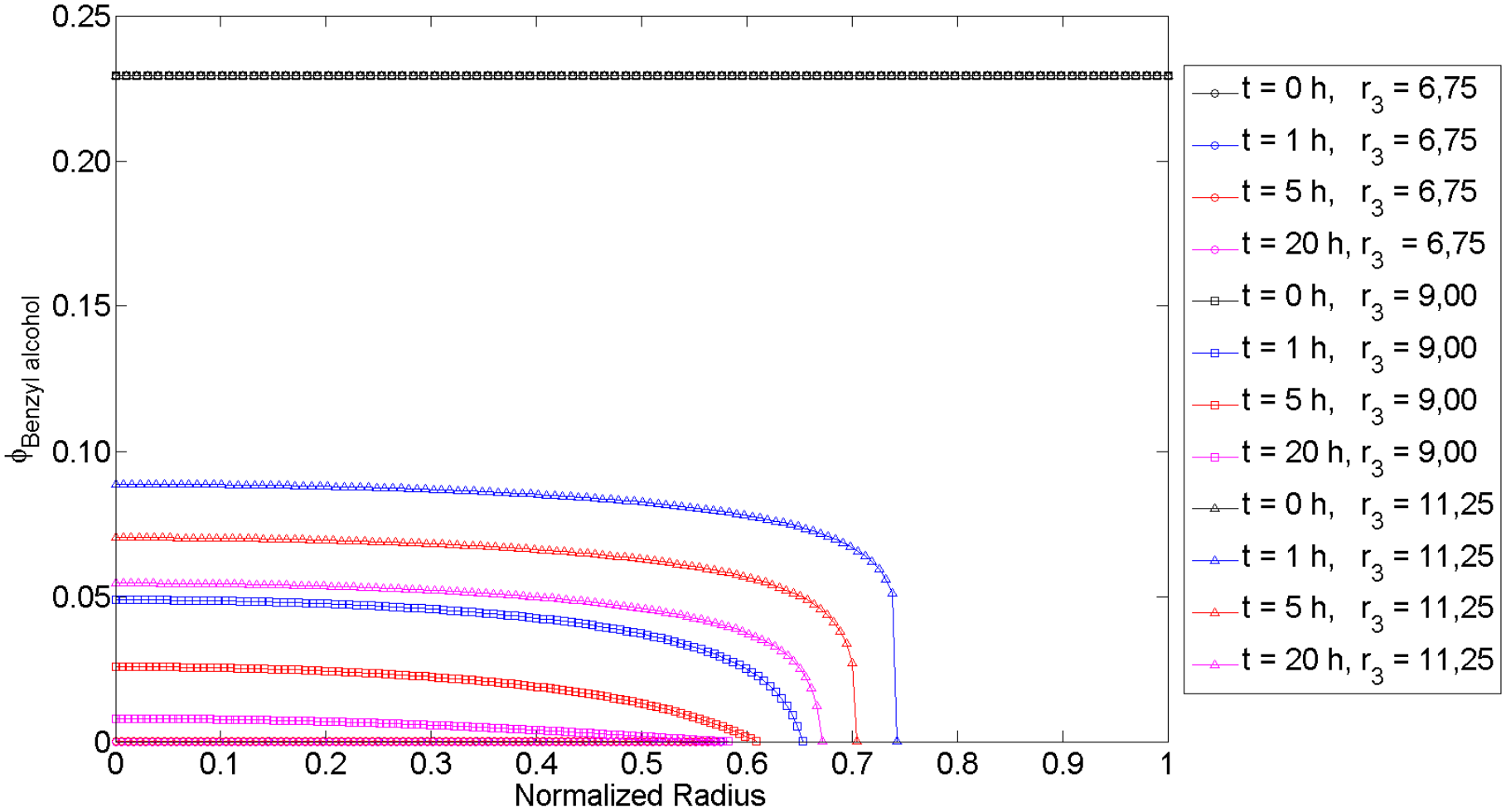



**Figure 6-12: Volume fraction of benzyl alcohol over the normalized radius for different values of $r_3$ at different instants of time.**

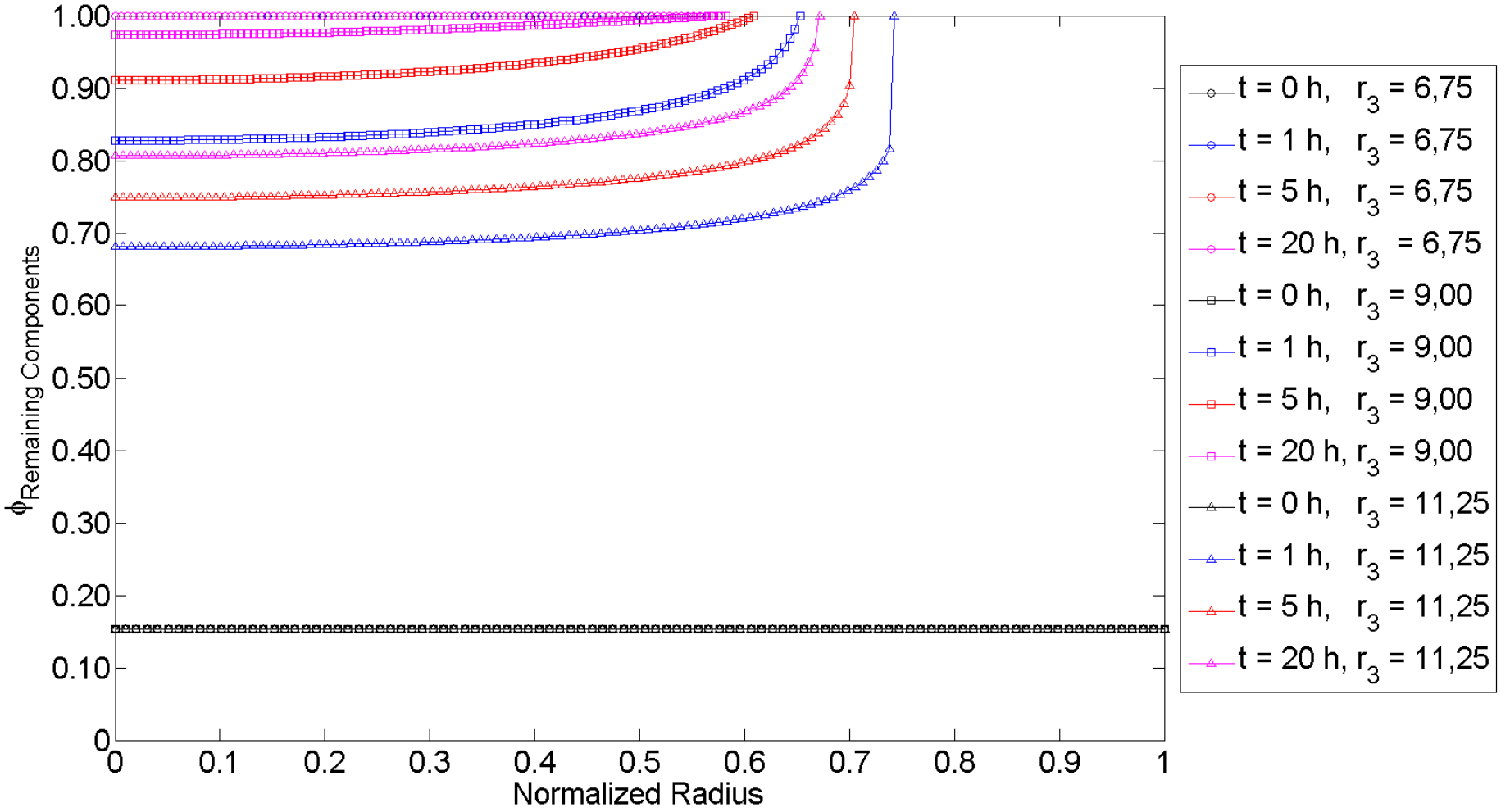



**Figure 6-13: Volume fraction of remaining components over the normalized radius for different values of $r_3$ at different instants of time.**

In Figure 6-14, Figure 6-15 and Figure 6-16 the diffusion coefficients of the two solvents with respect to the normalized radius for different values of $r_3$ at different instants of time are shown. The diffusion coefficients of the solvents change with the change of the volume fraction of polymer. Precisely:

- In Figure 6-14 $r_{2,EA}$ and $r_{2,BA}$ are set to be equal to 6,25 and $r_{3,EA}$ and $r_{3,BA}$ are set to be equal to 6,75.
- In Figure 6-16 $r_{2,EA}$ and $r_{2,BA}$ are set to be equal to 6,00 and $r_{3,EA}$and $r_{3,BA}$ are set to be equal to 9,00.
- In Figure 6-15 $r_{2,EA}$ and $r_{2,BA}$ are set to be equal to 5,75 and $r_{3,EA}$ and $r_{3,BA}$ are set to be equal to 11,25.

As shown in the figures the trend of the diffusion coefficients of the two solvents are equal. This is reasonable considering that the system-dependent constants of these two solvents were set to the same values. To validate this behavior further experimental analysis is required.

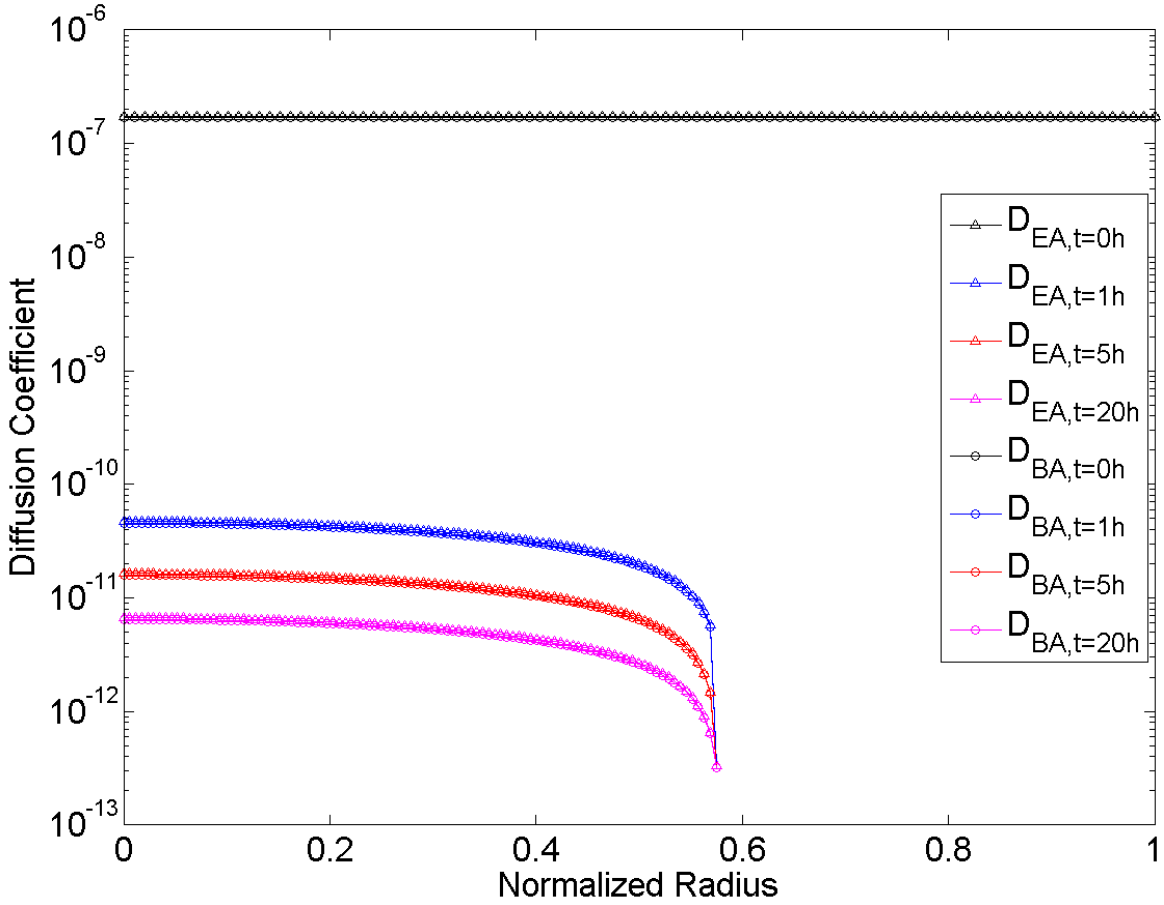



**Figure 6-14: Diffusion coefficients of the two solvents over the normalized radius for different extraction times.**

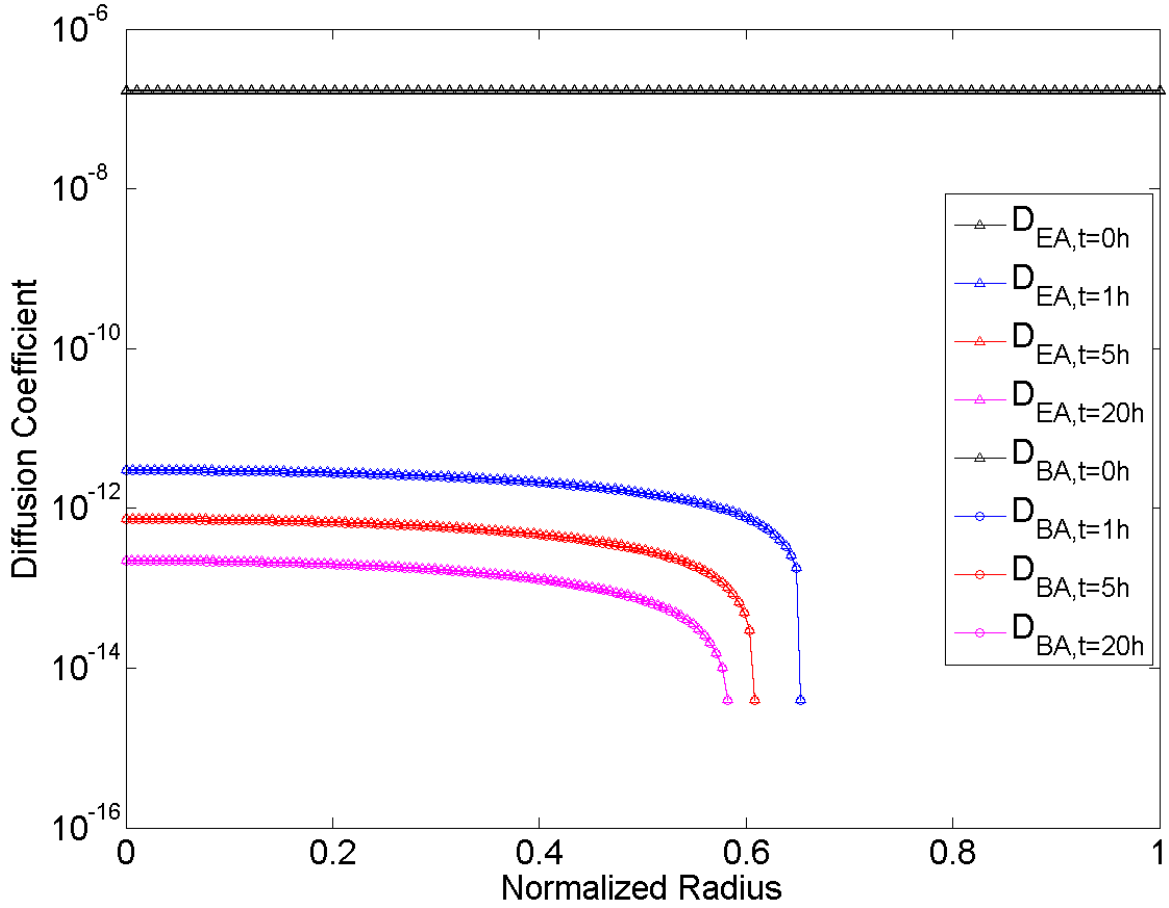



**Figure 6-16: Diffusion coefficients of the two solvents over the normalized radius for different extraction times.**

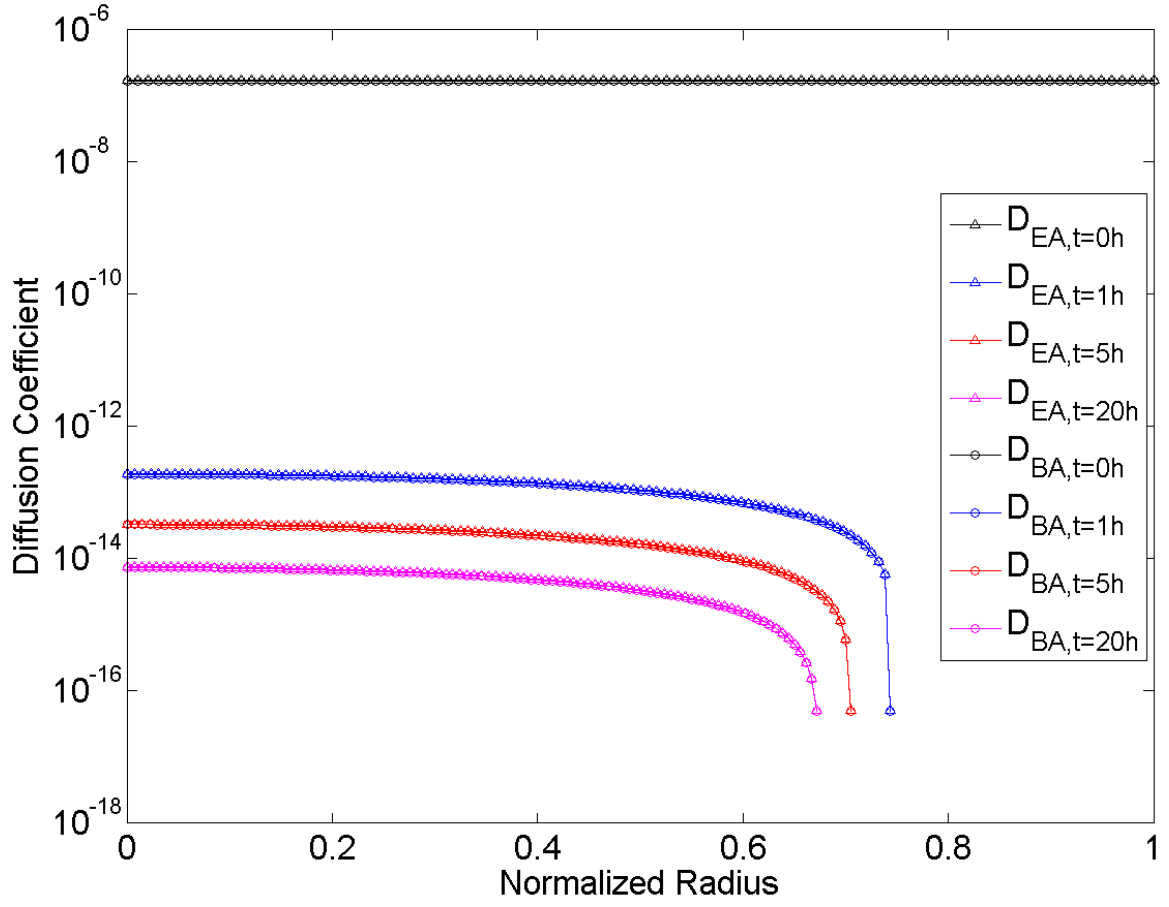



**Figure 6-15: Diffusion coefficients of the two solvents over the normalized radius for different extraction times.**

Finally a surface plot for the volume fraction of each component as a function of time and radius are shown in Figure 6-17, Figure 6-18 and Figure 6-19. These diagrams relate to the best match between numerical and experimental data, namely with $r_{3,EA}$ and $r_{3,BA}$ of the two solvents equal to 9, as well as $r_{2,EA}$ and $r_{2,BA}$ equal to 6.

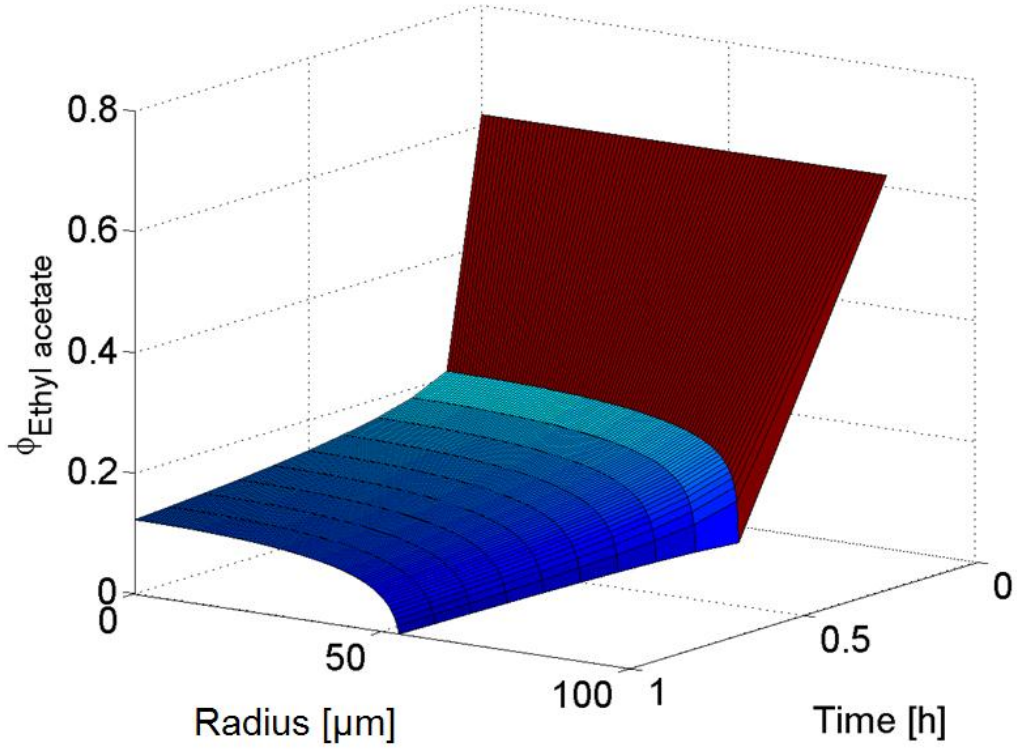



**Figure 6-17: Volume fraction of ethyl acetate as a function of the time and radius.**

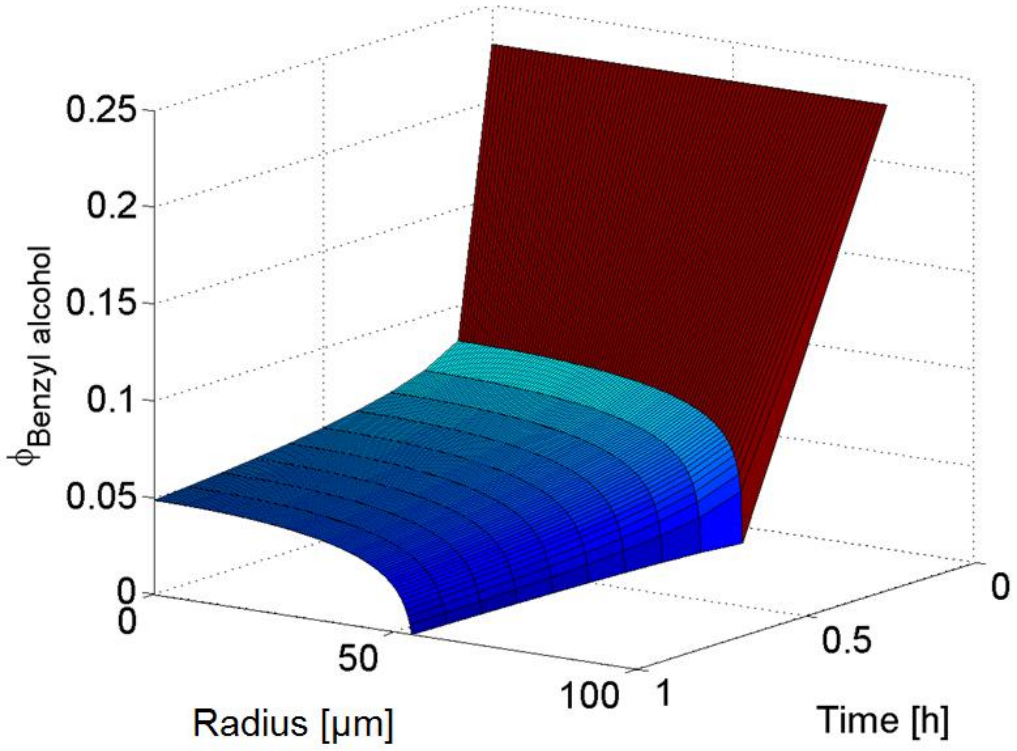



**Figure 6-18: Volume fraction of benzyl alcohol as a function of the time and radius.**

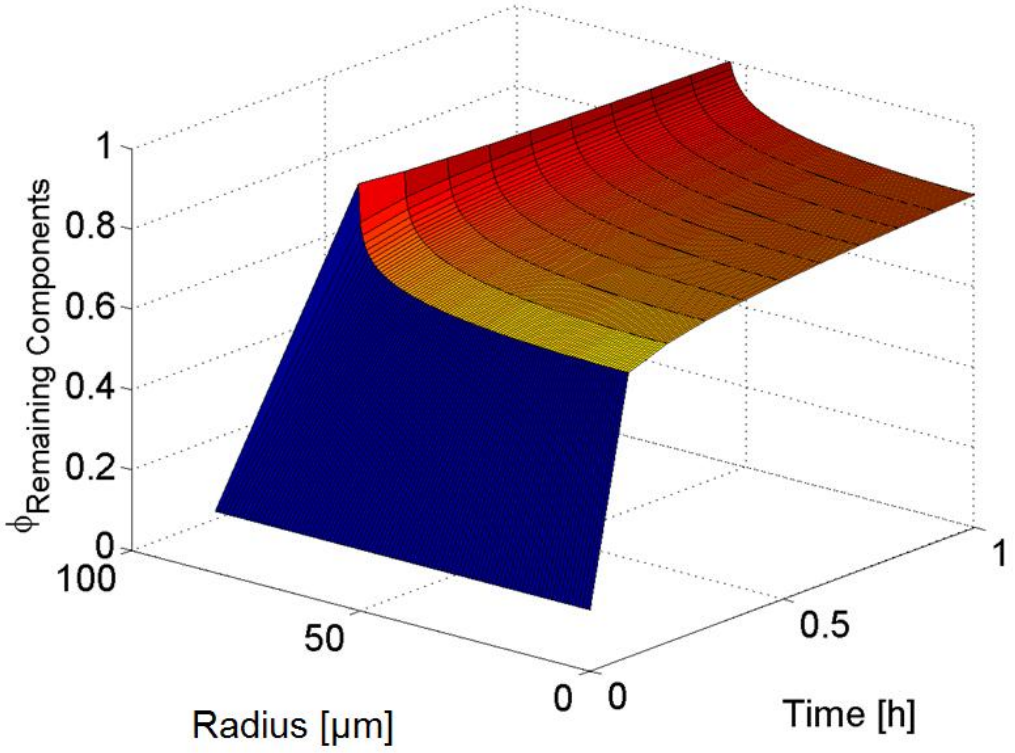



**Figure 6-19: Volume fraction of the remaining components (API, Polymer) as a function of the time and radius.**

It is evident that the volume fraction of the ethyl acetate and benzyl alcohol decrease fast, at the beginning, until the outer shells are almost completely free from solvent. The low diffusion coefficient inside the polymer leads then to a slow mass transport from the middle to the outer part of the micro-particle, thus to a slower extraction.

## 6.6. Diameter Variation Analysis

As described in Chapter 3 the emulsion was produced by a static mixer unit and was then led into the extraction reactor. It was determined that the emulsion drop size distribution depended on different parameters like volume flow and number of SMX static mixer elements. Consequently, the developed model had to be able to deal with different initial values for the emulsion drop size.

The investigated range of the initial diameter (t=0) variations are shown in Table 6-17. The same simulation parameters described in the previous sections, were used. Furthermore, the investigation was performed with the empirical parameters $r_i$ that provide the best agreement between experimental and numerical data. Hence, the most important simulation parameters are shown in Table 6-14 and the system values of the system dependent constants are presented in Table 6-15 and Table 6-16.

**Table 6-17: Investigated range of initial diameters.**

| Case | Diameter at the beginning $d_{t=0}$ [µm] |
|---|---|
| 1 | 50 |
| 2 | 100 |
| 3 | 163,49 |
| 4 | 200 |

The SMD over the extraction time for the different initial diameters are illustrated in Figure 6-20. It is apparent that the shrinkage depends on the initial size of the emulsion drops. To verify this observation, the ratio between the current diameter and the initial one was calculated as well. This percentage of shrinkage is shown over the extraction time in Figure 6-21.

This behavior could be easily explained as by decreasing micro-particle diameter the surface to volume ratio increases, leading to a stronger mass transfer from the surface to the extraction medium and, consequently, to an increase in the percentage of shrinkage. This ratio can be calculated by using Eqn. (6-70) to Eqn. (6-72).

$$V_P = \frac{{d_P}^3 \cdot \pi}{6} \tag{6-70}$$

$$A_P = {d_P}^2 \cdot \pi \tag{6-71}$$

$$\frac{A_P}{V_P} = \frac{6}{d_p} \tag{6-72}$$

Here $V_p$ is the volume, $d_p$ is the diameter and $A_p$ is the surface of the micro-particle.

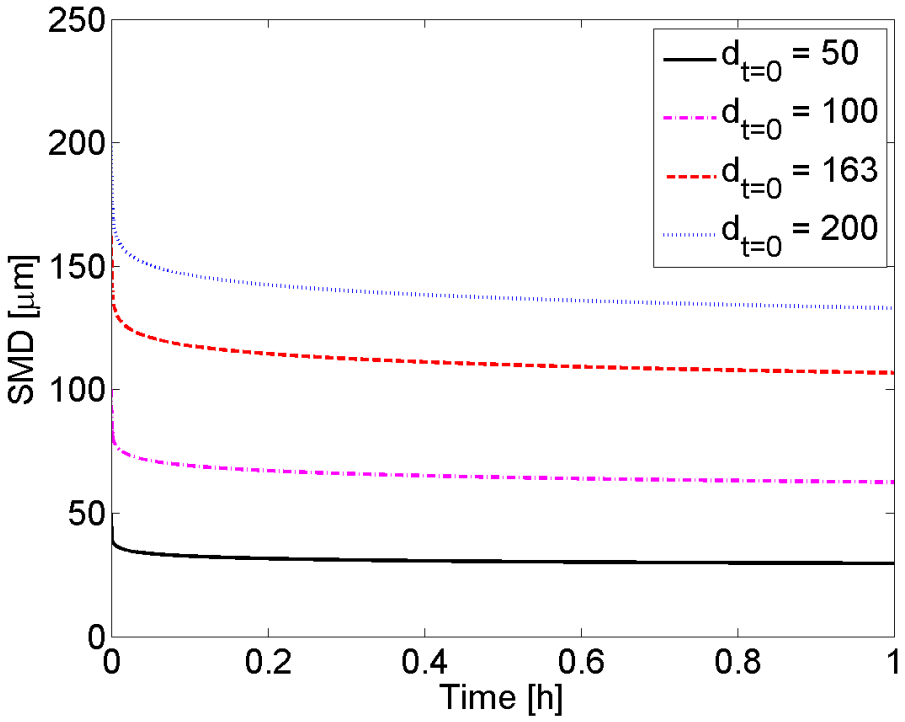



**Figure 6-20: SMD over time for different initial diameters.**

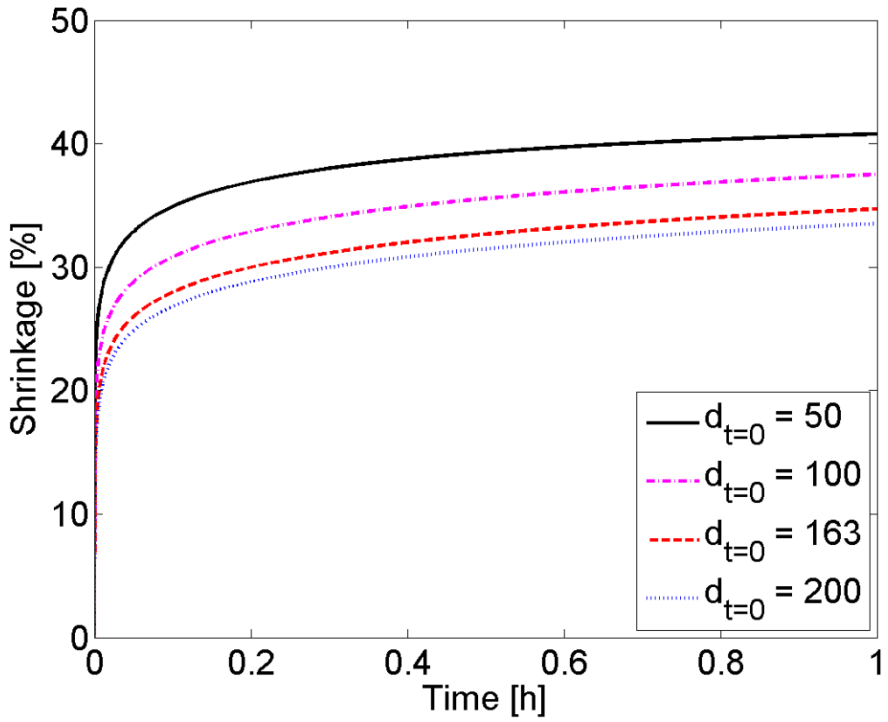



**Figure 6-21: Ratio of difference between the current and the initial diameter.**

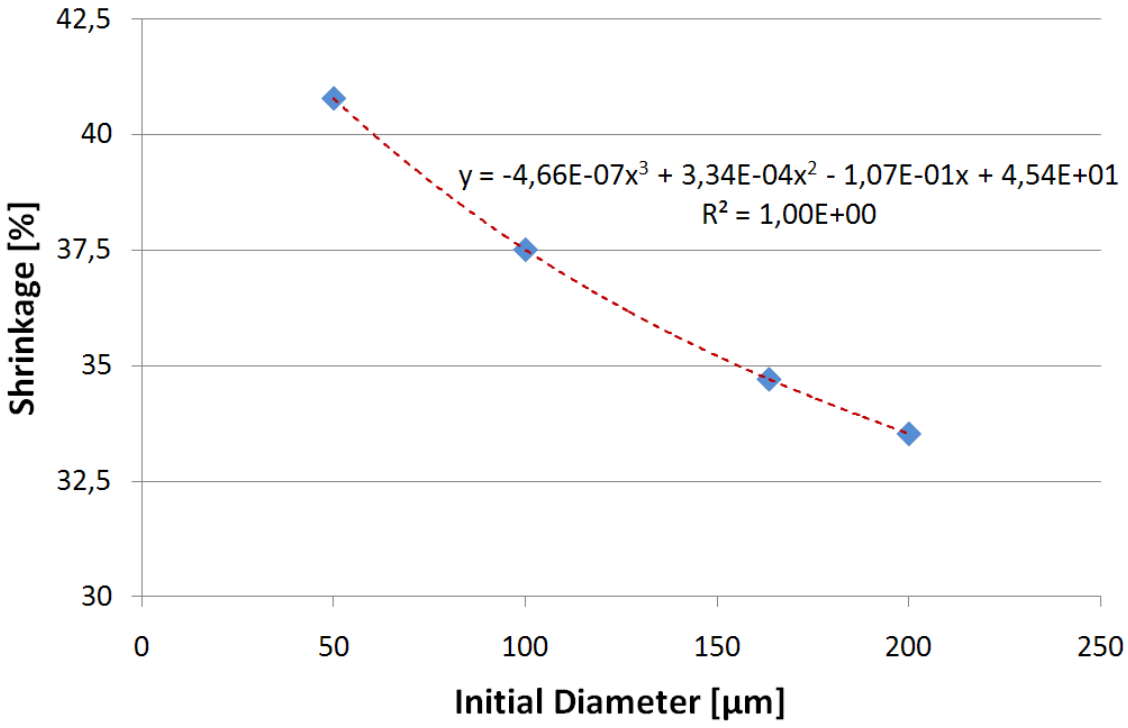



**Figure 6-22: Shrinkage over the different initial diameters.**

In Figure 6-22 the Shrinkage in percent over the initial diameter is illustrated. In the investigated range of the micro-particle size from 50 to 200 µm and with a revolution rate

of the stirrer of 260 rpm, the shrinkage ratio is almost linearly correlated to the initial diameter as the 2nd and 3rd exponent of the series are very low.

In Table 6-18 the most important results of the diameter variation analysis are summarized again.

**Table 6-18: Results of the diameter variation analysis.**

| Case | Diameter at the beginning $d_{t=0}$ [µm] | Diameter after 1 hour $d_{t=1h}$ [µm] | Shrinkage after 1 hour [%] |
|---|---|---|---|
| 1 | 50 | 29,61 | 40,78 |
| 2 | 100 | 62,49 | 37,51 |
| 3 | 163,49 | 106,75 | 34,71 |
| 4 | 200 | 132,95 | 33,53 |

## 6.7. Overview over the Experiments

The presented mass transfer model takes only the mean value for the relative velocity between particles and flow into account. To determine the influence of other parameters, like the local flow field, on the final micro-particle size, different experiments were conducted. For this purpose the revolution rate n and the position of the inlet point were varied. According to this, in Figure 6-23 a sketch of the reactor unit is shown where the position-determining variables $h_e$ and $x_e$ are defined. The engineering drawing of the reactor and stirrer unit can be found in the appendix. The different variations are summarized in Table 6-19.

**Table 6-19: Revolution rate n and distance between the inlet position and stirrer blade for the different experimental cases.**

| Cases | Revolution rate n [rpm] | Distance between inlet position and stirrer blade $h_e$ [mm] |
|---|---|---|
| 1 | 100 | 30 |
| 2 | 260 | 100 |
| 3 | 260 | 30 |

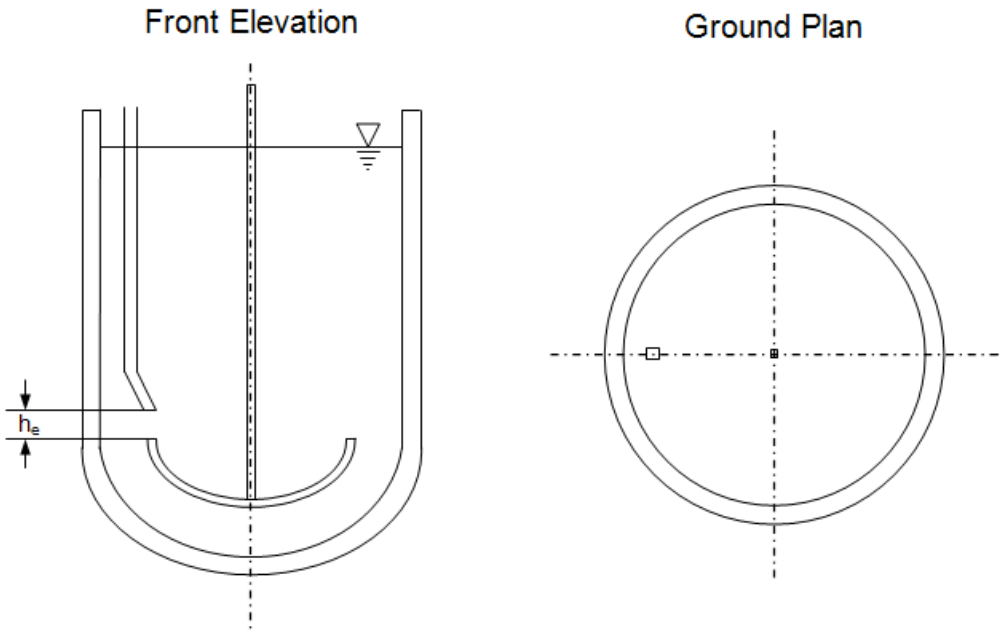



**Figure 6-23: Sketch of the extraction reactor to define the inlet position.**

## 6.8. Experimental Results

In Figure 6-24 the Sauter mean diameter (SMD) is shown over the extraction time for the three investigated cases.

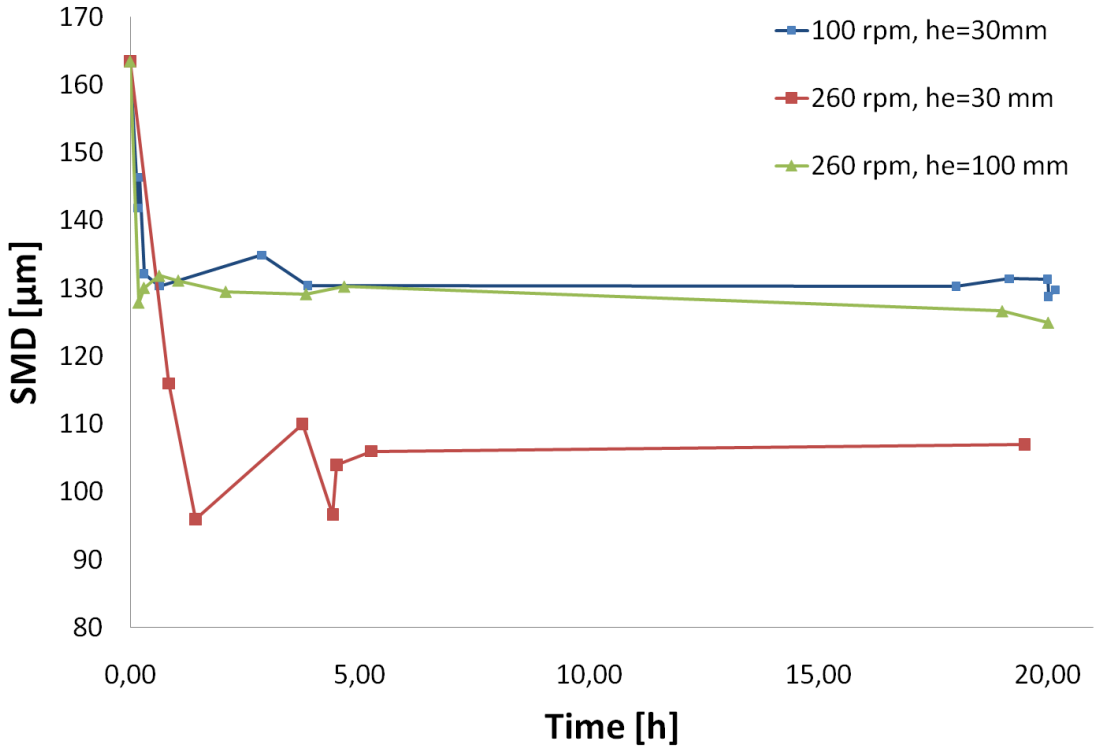



**Figure 6-24: Experimental results for the diameter of the micro-particle over time during extraction process for the three variations.**

As shown in the figure the revolution rate and the position of the inlet point have an influence on the final micro-particle size.

In case 1 where the revolution rate was set to 100 rpm and the distance between the inlet point and the stirrer blade was about 30 mm, represented by the blue line, the micro-particle diameter after twenty hours of extraction reduced from 163,49 µm to about 130 µm. This low revolution rate led to a lower local Reynolds number (see Equation (5.22)) respectively to a lower local velocity at the inlet point.

A complete different behaviour was observed when the revolution rate was set to 260 rpm and the distance between the inlet point and the stirrer blade was about 30 mm. In this so called standard case, represented by the red line, the local Reynolds number and the local velocity at the inlet point were much higher than in case 1. The micro-particle shrunk much more during the 20 hours of extraction to a final diameter of about 107 µm.

As a consequence of these facts it was assumed that the final micro-particle diameter can be related to the value of the local Reynolds number and the local velocity at the inlet point.

To determine the influence of the position of the inlet point the revolution rate was set to 260 rpm and only the distance between the inlet point and the stirrer blade was increased to about 100 mm. This larger pitch had the effect that the emulsion was inserted in an area with lower local velocity. After 20 hours of extraction the micro-particle diameter was around 130 µm. On the whole the observed trend was very similar to the case 1 where the revolution rate was much lower.

This additional insight suggests that the prevailing flow regime respectively the local velocity at the point of inlet is of extraordinary significance for the entire process.

The volume-based size and density distributions of the three cases after 20 hours of extraction are shown in Figure 6-25, Figure 6-26 and Figure 6-27.

In the cases where the emulsion was inserted in an area of low local velocity, case 1 and 2, the median value $x_{50}$ was around 200 µm. Furthermore the density distribution was a little bit narrower than in case 3, where the local velocity at the point of inlet was much higher.

To sum up, the following considerations can be derived from experimental results:

- The position of the inlet point is very important because the prevailing flow regimes at this point have a great influence on the size distribution of the micro-particles.
- If the emulsion is inserted in an area of low local velocity the resulting micro-particles are larger than the one produced at higher stirrer speed.
- The emulsions size distribution appears tighter for emulsions injected in regions of lower local Reynolds numbers.

Further research is nevertheless required to ensure the observations.

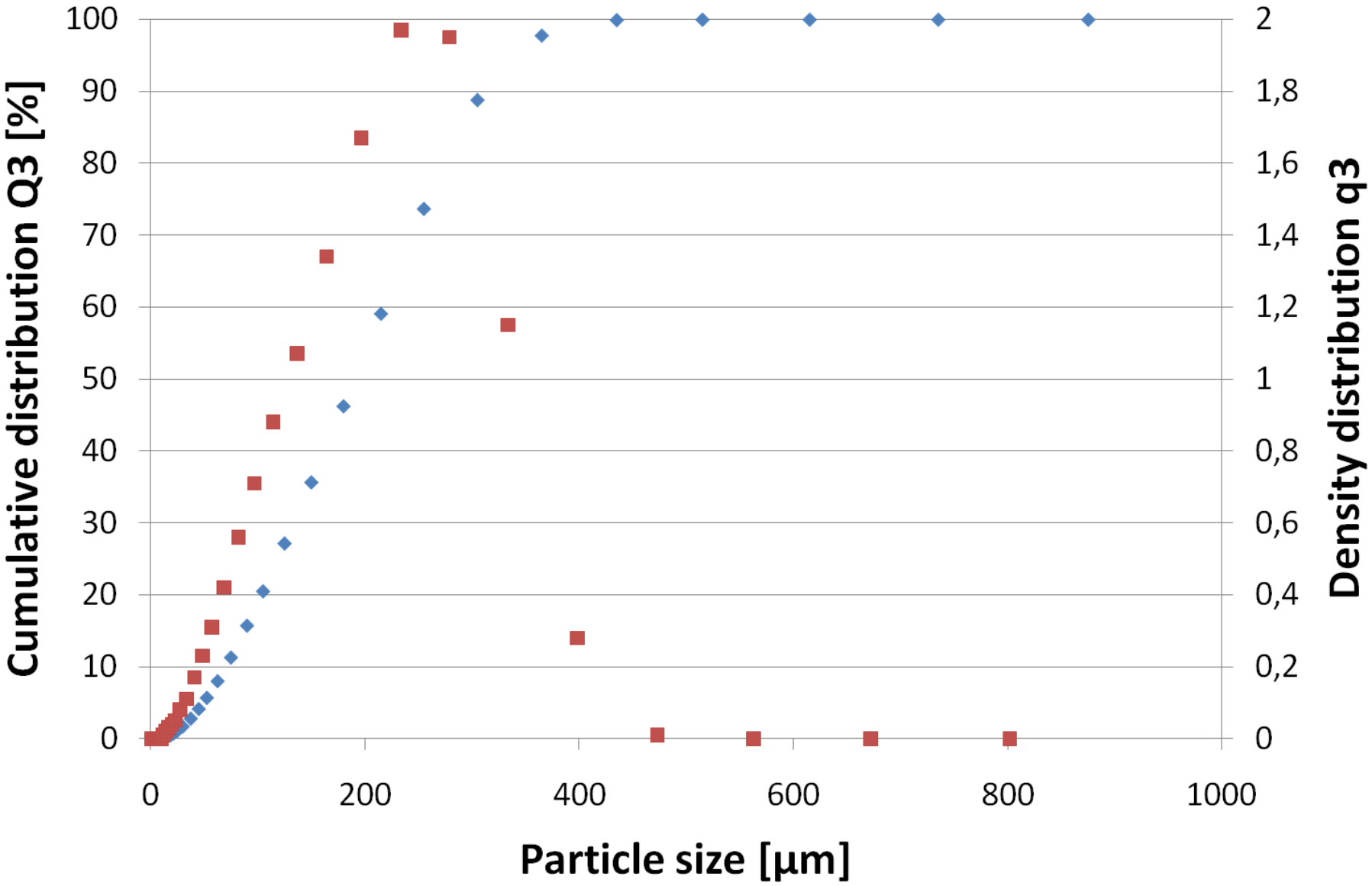



**Figure 6-25: Experimental results for case 1 after 20 hours of extraction (n = 100 [rpm] and $h_e$ = 30[mm]).**

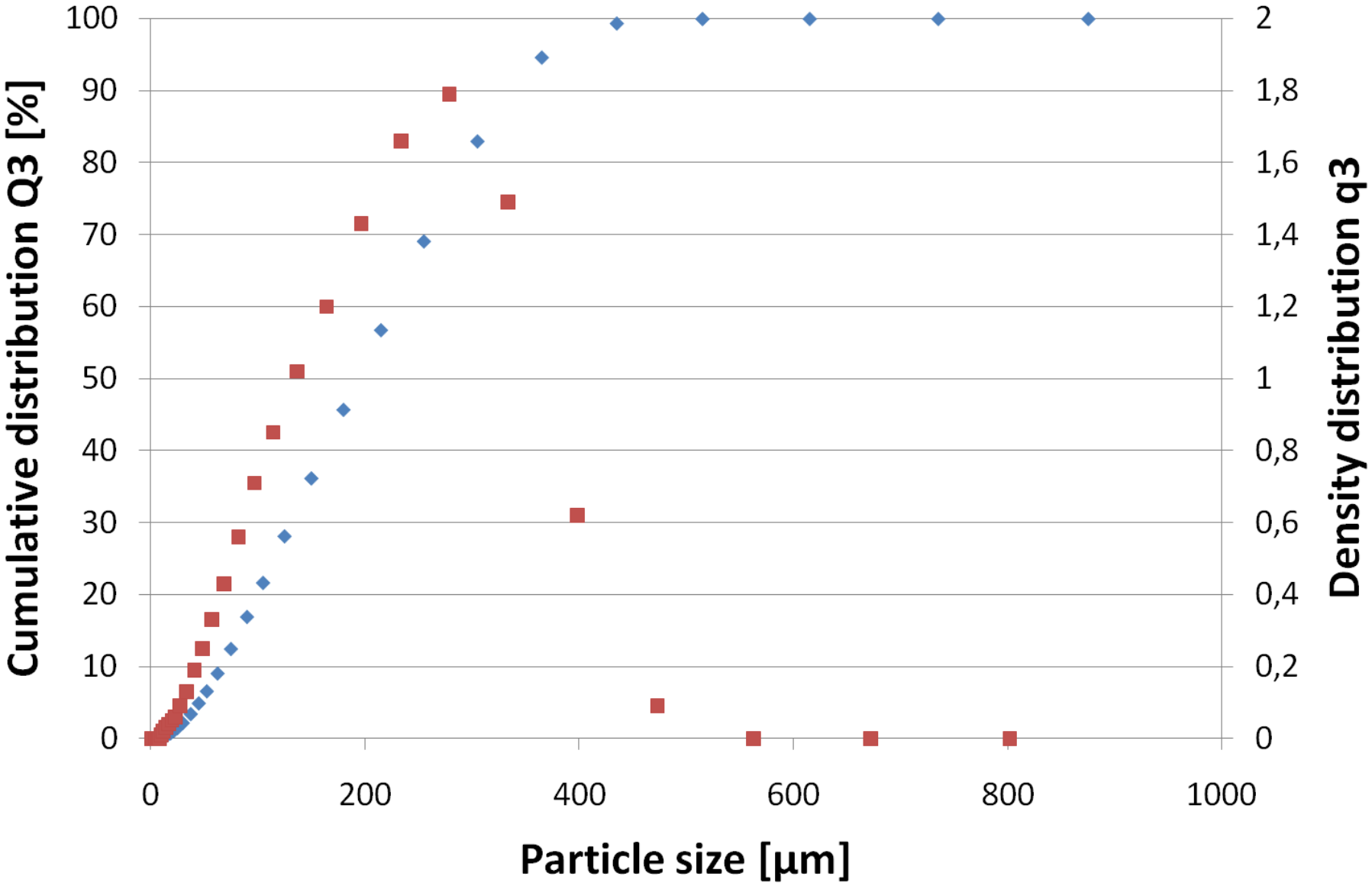



**Figure 6-26: Experimental results for case 2 after 20 hours of extraction (n = 260 [rpm] and $h_e$ = 100[mm]).**

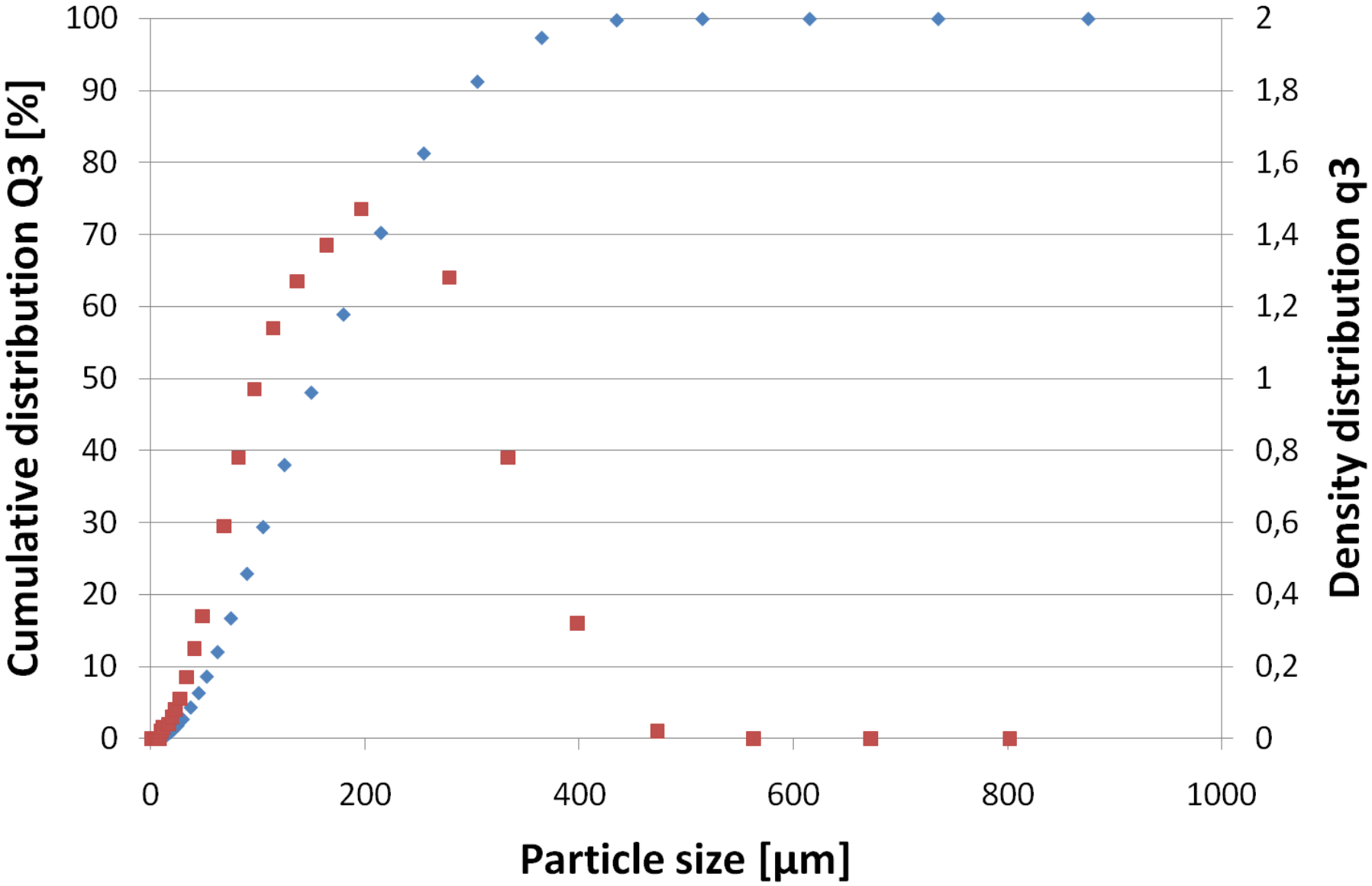



**Figure 6-27: Experimental results for case 3 after 20 hours of extraction (n = 260 [rpm] and $h_e$ = 30[mm]).**

# 7. Conclusions and Outlook

## 7.1. Conclusions

The objective of this work was to investigate the extraction step of a micro-particle process on the basis of solvent extraction method to predict different characteristics and to determine the most important process parameters. The final quality of the considered micro-particles can be controlled only by a deep understanding of each step of the production process. Therefore on the one hand a numerical model was developed and on the other hand different experimental investigations were performed.

The developed numerical model is capable to describe the mass transfer from the disperse phase into the continuous phase respectively the hardening/extraction process. First a mass transport model based on the work of Li et al. [18] was established using mass diffusion theory and thermodynamic principles. Therefore the different phenomena and variables, like the mass transport coefficient and the concentration dependent diffusion coefficient, were described and modeled mathematically. Also the comparison of the obtained numerical results by this model with an analytical solution, found by John Crank [7] showed a good agreement. Furthermore a custom build shrinking model respectively a method to numerically compute the diameter reduction of the micro-particles was implemented. Unfortunately, assumptions had to be made in order to perform the simulation due the fact that different data on chemical media are not exactly known and occurring phenomena are not fully researched.

However predictions of our numerical model fit the experimental data reasonably well. Therefore the experimental obtained results of the micro-particle size were compared with the numerical ones. Besides that to determine the discretization uncertainty the influence of the time step size and the grid size on the accuracy of the results were also investigated. In order to find realistic ranges for the values of the unknown variables a parameter study was performed. The results of the sensitivity analyses indentified the correlation of the concentration depending diffusion coefficient respectably the system-dependent constants as the most influential parameters. Consequently these parameters are well suited to adjust the results of the simulation.

The numerical model clearly showed that the mass transport inside the micro-particle is the determining step in the extraction process. It is much slower than the mass transport from the interface to the continuous phase. As a result the limiting step of the extraction process is given by the diffusion in the polymer.

Concerning the experimental investigations first of all an experimental set up was build up in the laboratories of the Research Center Pharmaceutical Engineering (RCPE GmbH) in Graz. The performed investigations were able to quantify the micro-particle size during the extraction process. Therefore two particle size analysis systems from Sympatec GmbH., namely HELOS and QICPIC were used. Also the influences of different parameters, like the revolution rate of the stirrer and the position of the inlet point, on the final micro-particle size distribution were investigated.

These investigations suggested that the first seconds of the process have the greatest influence on the final "quality" of the micro-particles. It was assumed that the prevailing flow regimes, at the beginning this means at the first contact of disperse and continuous phase in the extraction tank determine the final properties of the micro-particles.

On the whole I have to mention that most of the goals set in this work were reached. The developed numerical model is a first approach to get a better understanding of the extraction stage. It also helps to determine the most important process parameters and which influence they have on the final micro-particle quality.

To sum up the gained knowledge provides a very useful guideline for the process technology development. This helps on the one hand to reduce the tedious trial and error experiments and on the other hand to cut development costs.

Finally I have to mention that a better process understanding is the key to superior and more cost efficient pharmaceuticals.

## 7.2. Future Work

Further investigations, both experimental and numerical, of the observed processes are of utmost importance. For example, numerical CFD (Computational Fluid Dynamics) simulations of the flow field inside the reactor may be useful to understand the local effects on the hardening process. These computations can provide information about the prevailing flow regimes in the reactor and can lead to a better insight here. In addition to this, the developed numerical MATLAB model should be linked with the CFD simulations. The gained information should then be approved experimentally.
Moreover, microscopy measurements of the micro-particle during the extraction process could provide a deeper understanding on the structure of the hardening polymer.

Another future task is the further improvement and testing of the numerical model. Here, the different unknown parameters have to be determined whether by experimental investigations or by further literature research. These new information should then be implemented in the existing model. Besides all that, a variation the revolution rate of the stirrer should be investigated.

Further experimental validations of the numerical results should be also carried out to verify the model in terms of mass transfer. To achieve this, the volume fraction of the solvents in the continuous phase during the extraction process should be determined. The entire required chemical analysis (GC method) was already established and tested during this work.

Another future task can be the evaluation of different measurement systems for the inline/online/real-time monitoring of the process. The potential of these techniques should be assessed in terms of accuracy and measuring speed. A previous analysis of measurement systems, performed during this work, showed that spectroscopic systems (e.g. Raman-, Near-infrared- and Mid-infrared-Spectroscopy) fulfill most of the requirements and are therefore the best possibilities to measure the volume fractions of the solvents (ethyl acetate and benzyl alcohol) in the continuous phase.

# 8. List of References

[1] The MathWorks, Inc., MATLAB Product Help. Natick (2010).

[2] Baehr, H. D.; Stephan, K. Wärme- und Stoffübertragung. Second Edition. Springer-Verlag Berlin Heidelberg New York, (1996).

[3] Bakken, K.; Heruth, E. E.: Temporal Control of Drugs. Annals of the New York Academy of Sciences (618) (1991), 422–427.

[4] Bird, R. B.; Stewart, W. E. Lightfoot, E. N. Transport Phenomena. Revised Second Edition. John Wiley & Sons Inc, New York (2007).

[5] Birnbaum, D. T.; Brannon-Peppas, L.: Microparticle Drug Delivery Systems in Drug Delivery Systems in Cancer Therapy. Humana Press Inc. (2004).

[6] Lewy, R.; Courant, R.; Friedrichs, R.: On the partial Difference Equations of Mathematical Physics. IBM JOURNAL (1967), 215–234.

[7] Crank, J. The Mathematics of Diffusion. Second Edition. Clarendon Press, Oxford (1975).

[8] Flory, P. J. Principles of Polymer Chemistry. Cornell University Press, New York (1953).

[9] Freitas, S.; Merkle, H. P.; Gander, B.: Microencapsulation by Solvent Extraction/Evaporation: Reviewing the State of the Art of Microsphere Preparation Process Technology. Journal of Controlled Release (102) (2005), 313–332.

[10] Freytag, T.; Dashevsky, A.; Tillman, L.; Hardee, G.E.; Bodmeier, R.: Improvement of the Encapsulation Efficiency of Oligonucleotide-Containing biodegradable Microspheres. Journal of Controlled Release (69) (2000), 197–207.

[11] Gandera B.; Blanco-Prìeto, M. J.; Thomasin, C.; Wandrey, Ch.; Hunkeler, D.: Coacervation and Phase Separation. Encyclopedia of Pharmaceutical Technology (3) (2006).

[12] Grenville, K. R.; Nienow, W. A. Blending in the laminar Regime; in: Handbook of Industrial Mixing. John Wiley & Sons, Inc., Hoboken, New Jersey (2004).

[13] Herrmann, J.; Bodmeier, R.: Biodegradable, Somatostatin acetate containing Microspheres prepared by various aqueous and non-aqueous Solvent Evaporation Methods. European Journal of Pharmaceutics and Biopharmaceutics (45) (1998), 75–82.

[14] Higham, D. J.; Highman, N.J. MATLAB Guide. Society for Industrial and Applied Mathematics, Philadelphia (2000).

[15] Izumikawa, S.; Yoshioka, S.; Aso, Y.; Takeda, Y.: Preparation of poly(l-lactide) microspheres of different crystalline Morphology and Effect of crystalline Morphology on Drug Release Rate. Journal of Controlled Release (15) (1991), 133–140.

[16] John, E.; Hammerschmidt, W.; Paulus, K.; Petersen, H.: Process and Product Control of porous Microparticles - a long acting release Formulation. PARTEC 2007 (2007).

[17] Jyothi, N. V.; Prasanna, M.; Prabha, S.; Ramaiah, S. P.; Srawan, G.; Sakarkar, S. N.: Microencapsulation Techniques, Factors influencing Encapsulation Efficiency: A Review. The Internet Journal of Nanotechnology (3) (2009).

[18] Li, W. I.; Anderson, K. W.; DeLuca, P. P.: Kinetic and Thermodynamic Modeling of the Formation of polymeric Microspheres using Solvent Extraction/Evaporation Method. Journal of Controlled Release (37) (1995), 187–198.

[19] Li, M.; Rouaud, O.; Poncelet, D.: Microencapsulation by Solvent Evaporation: State of the Art for Process Engineering Approaches. International Journal of Pharmaceutics, (363) (2008), 26–39.

[20] Luan, X.: Biodegradable Microparticle and In Situ Microparticle Systems. PhD Thesis. Freien Universität Berlin (2005).

[21] Wilkinson, A.; McNaught, A. D. Compendium of Chemical Terminology, The Gold Book. Blackwell Science (1997).

[22] Kiss, N.; Brenn, G.; Pucher, H.; Wieser, J.; Scheler, S.; Suzzi, D.; Khinast, J.: Formation of o/w Emulsions by Static Mixers for Pharmaceutical Applications. Draft of a Publication for Chemical Engineering Science.

[23] Jèrrôme, R.; Teyssié, P.; Nihant, N.; Grandfils, C.: Microencapsulation by Coacervation of poly(lactide-co-glycolide) iv. Effect of the processing Parameters on Coacervation and Encapsulation. Journal of Controlled Release (35) (1995), 117–125.

[24] Putney, S. D.; Burke, P. A.: Improving protein Therapeutics with sustained-release Formulations. Nature biotechnology (16) (1998), 153–157.

[25] Reuvers, A. J.; Smolders, C. A.: Formation of membranes by means of immersion precipitation 2. The mechanism of formation of membranes prepared from the system cellulose-acetate acetone water. Journal of Membrane Science, (34) (1987), 67–86.

[26] Perry R.H. Perry's Chemical Engineers' Handbook, Volume 7. McGraw-Hill Book (1998).

[27] Santini, J.T.; Richard A. C.; Scheidt, R.; Cima, M. J.; Langer R.: Microchips as Controlled Drug-Delivery Devices. Angewandte Chemie International Edition (39) (2000), 2396–2407.

[28] S., Scheler. Polymermikropartikel; in Moderne Pharmazeutische Technologie (Keck, C. M.; Müller, R.H.) (2009).

[29] Schweizer, W. MATLAB kompakt. Second Edition. Oldenbourg Wissenschaftsverlag GmbH. München (2007).

[30] Taylor, R.; Krishna, R. Multicomponent Mass Transfer. John Wiley & Sons Inc., New York (1993).

[31] Tompa, H. Polymer Solutions. Academic Press Inc., New York (1956).

[32] Tosun, I. Modeling in Transport Phenomena, Second Edition. Elsevier B.V., Amsterdam (2007).

[33] Venkatesan, P.; Manavalan, R.; Valliappan, K.: Microencapsulation: A vital Technique in novel Drug Delivery System. Journal of Pharmceutical Sciences and Research (4) (2009), 26–35.

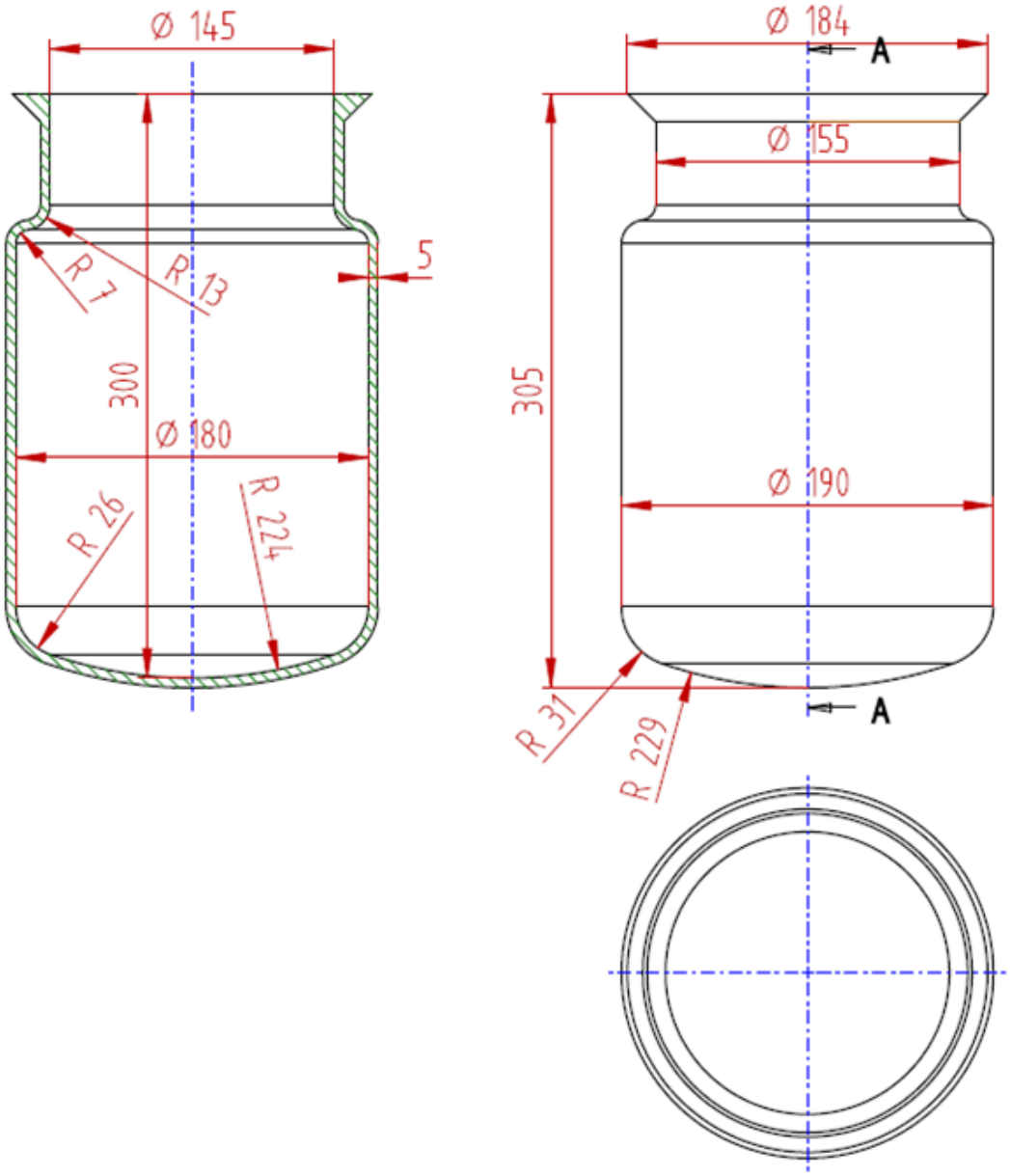
[34] VDI Gesellschaft Verfahrenstechnik und Chemieingenieurwesen Verein deutscher Ingenieure. VDI-Wärmeatlas. Springer Verlag Berlin Heidelberg (2006).

[35] Wang, J.; Schwendeman, S. P. Mechanisms of Solvent Evaporation Encapsulation Processes: Prediction of Solvent Evaporation Rate. Journal of Pharmaceutical Sciences 88(10) (1999), 1090–1099.

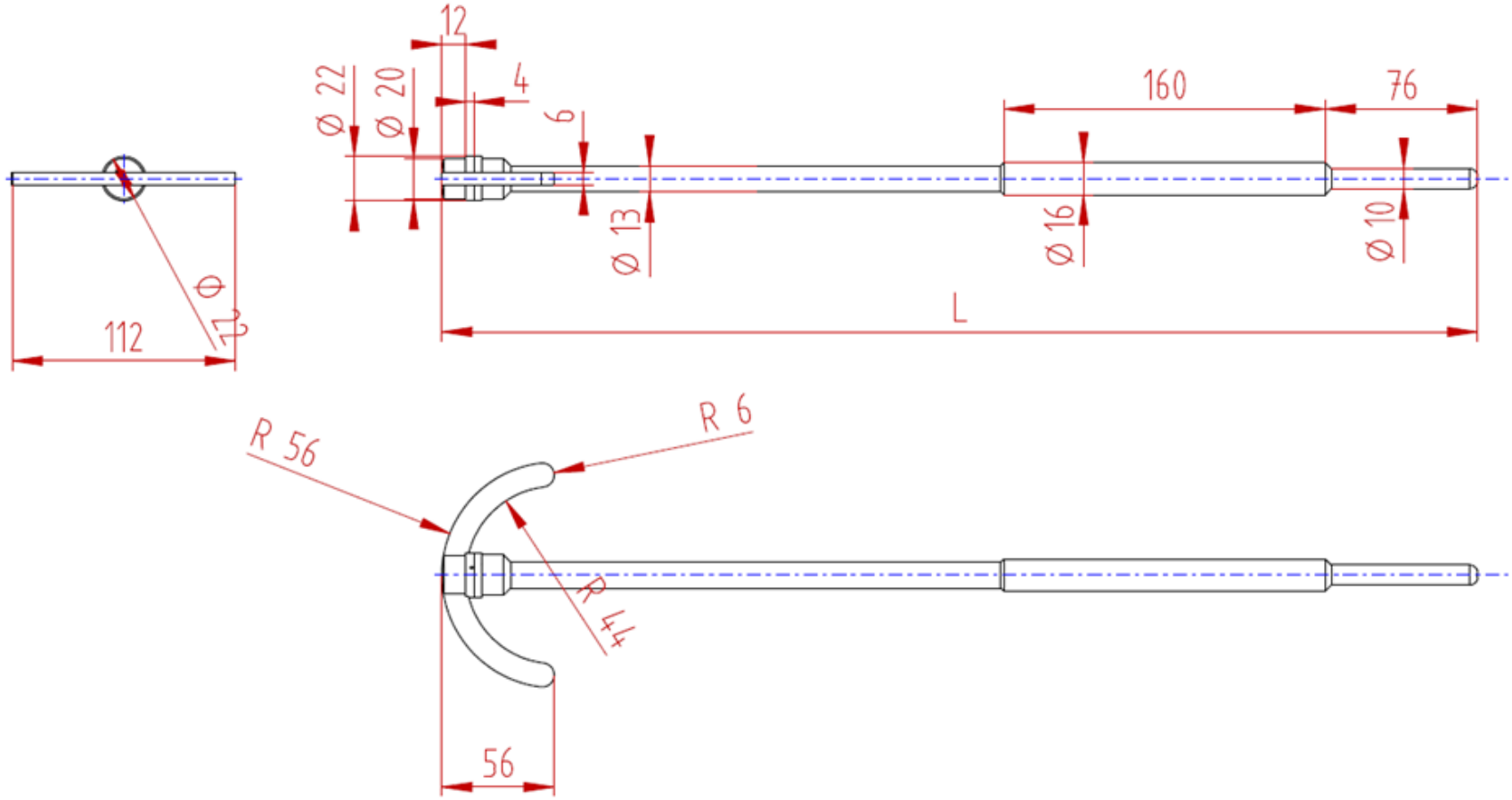
[36] Wilke, C. R.; Chang, P.: Correlation of Diffusion Coefficients in dilute Solutions. Aiche Journal, 1 (2) (1955), 264–270.

[37] N.N. HELOS, – http://www.sympatec.com/LaserDiffraction/HELOS.html – accessed 01.12.2010.

[38] N.N. QICPIC, – http://www.sympatec.com/ImageAnalysis/QICPIC.html – accessed 01.12.2010.




| | | | | | | | |
|---|---|---|---|---|---|---|---|
| (Range of application) | | | | (Adm. tol.) | | Surface | Scale 1:1 | (Weight) |
| | | | | | | | (Basic material)<br>(Raw part number)<br>(Modell or casting number) | |
| | | | | | Date | Name | Reactor | |
| | | | | Editor | | | | |
| | | | | Reviewer | 9.2.2010 | Pucher | | |
| | | | | Norm | | | | |
| | | | | | | | | Page |
| | | | | | | | | Pages |
| Auth. | Revision | Date | Name | Source | | | Replacement for: | |




| | | | |
|---|---|---|---|
| (Range of application) | (Adm. tol.) | Surface | Scale 1:1 (Weight) |
| | | | (Basic material)<br>(Raw part number)<br>(Modell or casting number) |

| | | | | | Date | Name |
|---|---|---|---|---|---|---|
| | | | | Editor | | |
| | | | | Reviewer | 9.2.2010 | Pucher |
| | | | | Norm | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| Auth. | Revision | Date | Name | Source | | |

Stirrer

Page

Pages

Replacement for:

# I. Code (MATLAB R2009b)

## Calc.m

```
% Numerical Model
% (c) by Hannes Pucher
%--------------------------------------

%--------------------------------------
% Clear, Close and delete all
clc
clear all
close all
%--------------------------------------

%--------------------------------------
% Stopwatch start

tStart=tic
%--------------------------------------

%--------------------------------------
% Load the different parameter files

cd
DiffusionCoefficient
Parameters
%--------------------------------------

%--------------------------------------
% Define the memory variables

memRadius_end = [];
memTime = [];
memVolumefraction_EA = [];
memVolumefraction_BA = [];
memRadius = [];
memMass = [];
memAmbient_EA = [];
memAmbient_BA = [];
membetafl_EA = [];
membetafl_BA = [];
memDiffusion = [];
%--------------------------------------

%--------------------------------------
% Solve the differential equation for every delta t as long as t_total <
% t_total_max

while t_total < t_total_max

    % Call the PDEPE solver

    sol                         =
    pdepe(m,@calcpde,@calcic_2,@calcbc,radius,t);
```

```
% Extract the solutions as volume_fraction_EA/BA/W

volume_fraction_EA       = sol(:,:,1);
volume_fraction_BA       = sol(:,:,2);
volume_fraction_W        = sol(:,:,3);

% Call the numerical model for shrinking

Shrinking

% Extract the different solutions

volume_f_current_EA      = volume_fraction_EA(t_span,:); % Extract
the last volumefraction profile of ethylacetate
volume_f_current_BA      = volume_fraction_BA(t_span,:); % Extract
the last volumefraction profile of benzylalcohol

delta_m_EA_temp          = sum_delta_m_EA;
delta_m_BA_temp          = sum_delta_m_BA;

radius                   = radius_new_cell; % [m] Define the new
radius of the sphere
radialPositions          = radius; % [m] Define the new radial
position of every volumefraction
Surface_particle         = Surface_particle_new; % [m^2] Define the
new surface of the sphere

mass_ethylacetate_q_temp = mass_ethylacetate_q_new;
mass_benzylalcohol_q_temp= mass_benzylalcohol_q_new;

t_total                  = t_total + delta_t % [s] Define the new
time (t_total)

memRadius_end            = [memRadius_end radius(r_intervall)]; %
[m]Save the actual radius of the sphere
memTime                  = [memTime t_total]; % [s] Save the actual
time
memVolumefraction_EA     = [memVolumefraction_EA
volume_f_current_EA'];
% Save the actual volume fraction vector of ethylacetate
memVolumefraction_BA     = [memVolumefraction_BA
volume_f_current_BA'];
% Save the actual volume fraction vector of benzylalcohol
memRadius                = [memRadius radius']; % [m] Save the
actual
radial vector
memMass                  = [memMass sum_mass']; % [kg] Save the
actual
mass of the sphere
memAmbient_EA            = [memAmbient_EA ambientconc_EA']; % Save
the
actual ambient volume fraction of ethylacetate
memAmbient_BA            = [memAmbient_BA ambientconc_BA']; % Save
the
actual ambient volume fraction of benzylalcohol
membetafl_EA             = [membetafl_EA betafl_EA]; % [m/s] Save
the
actual mass transport coefficient of ethylacetate
membetafl_BA             = [membetafl_BA betafl_BA]; % [m/s] Save
the
```

```
    actual mass transport coefficient of benzylalcohol

end
%---------------------------------------
%---------------------------------------
% Plot file for the radius
Plot_Radius

% Plot file for the diffusion coefficients
Plot_Diffusion_EA
Plot_Diffusion_BA

% Plot file for the result for ethylacetate, benzylalcohol and water
Plot_EA
Plot_BA
%Plot_W

% 3D Plot of the volumefraction
SurfPlot

% Stopwatch stop
 tElapsed = toc(tStart)
%---------------------------------------
```

## Parameters.m

```
% Numerical Model Parameterfile
% (c) by Hannes Pucher
% Sources : Perry's handbook of chemical engineering, VDI, infosticker on
% the bottle of the chemicals
%--------------------------------------


% Definition of the global parameters

global m
global volume_f_initial_EA
global volume_f_initial_BA
global volume_f_initial_W
global volume_f_initial_API
global betafl_EA
global betafl_BA
global betafl_W
global ambientconc_EA
global ambientconc_BA
global ambientconc_W
global diffusion_coefficent_EA
global diffusion_coefficent_BA
global diffusion_coefficent_W
global radius
global t
global diffusion_c_EA_begin
global diffusion_c_EA_end
global r_EA_1
global r_EA_2
global r_EA_3
global diffusion_c_BA_begin
global diffusion_c_BA_end
global r_BA_1
global r_BA_2
global r_BA_3
global radialPositions
global volume_f_current_EA
global volume_f_current_BA
global Surface_particle
global memRadius_plot
global memVolumefraction_EA_plot
global memVolumefraction_BA_plot
global number_of_particles_m
%--------------------------------------

%--------------------------------------
% Variables for the solver

% Radius
d_measured                    = 163.49*10^-6; % [m] Diameter of the
emulsion measured by Msc. Nikolett Kiss
r_min                         = 0; % [m] Initial value for the radial
vector
r_max                         = d_measured/2; % [m] Final value for the
radial vector
r_intervall                   = 100; % Radial steps
```

```
radius                          = linspace(r_min,r_max,r_intervall); % [m]
Radial vector

% Time
t_total_max                     = 10; % [s] End time

t_total                         = 0; % [s] Total time
delta_t                         = 1; % [s] Time steps in wich the PDE will
be solved

t_min                           = 0; % [s] Initial value for the time
vector
t_max                           = delta_t; % [s] Final value for the time
vector
t_span                          = r_intervall; % Time steps

t                               = linspace(t_min,t_max,t_span); % [s] Time
vector
%--------------------------------------

%--------------------------------------
% Define the key parameters to get the radius, the volume fraction and
% every other parameter for the plot function at tau = 0.1

t_plot_intervall                = 0.1*t_total_max; % [s] Plot intervall for
tau = 0.1
t_intervall                     = t_total_max/t_plot_intervall; % [s]
Intervall for the plot function; get value every .. [s]
value_get                       = t_plot_intervall/delta_t; % Defines at
wich row the data for the plot will be stand in the matrix
%--------------------------------------

%--------------------------------------
% Parameters of the different components and solutions

density_quench                  = 0.9993; % [g/cm^3] Density of the quench
Solution measured by Msc. Nicolett Kiss
density_disp                    = 0.9948; % [g/cm^3] Density of the organic
phase measured by Msc. Nicolett Kiss
density_disp_vector             =
linspace(density_disp,density_disp,r_intervall-1); % [g/cm^3] Density
vector at the beginning (t=0)

% Water
density_water_5                 = 1.000; % [g/cm^3] Density of water at 5
[°C]
density_water_25                = 0.997; % [g/cm^3] Density of water at 25
[°C]
density_water                   = (density_water_5 + density_water_25)/2; %
[g/ml] Average density of ethylacetate

% Ethylacetate
density_ea_5                    = 0.9168; % [g/cm^3] Density of
ethylacetate at 5 [°C]
density_ea_25                   = 0.89362; % [g/cm^3] Density of
ethylacetate at 25 [°C]
density_ea                      = (density_ea_5 + density_ea_25)/2; %
[g/ml] Average density of ethylacetate
```

```
% Benzylalcohol
density_ba_20                   = 1.043; % [g/cm^3] Density of
benzylalcohol at 20 [°C]
density_ba                      = density_ba_20;% [g/cm^3] Average density
of benzylalcohol

% Polyvinylalcohol
density_pva                     = 1.250; % [g/cm^3] Density of the
polyvinylalcohol (NOT REAL VALUE)

% Polymer
density_polymer                 = 1; % [g/cm^3] Density of the polymer (NOT
REAL VALUE)

% API
%density_api                     = ?; % [g/cm^3] Density of the api,
definition follows (NOT REAL VALUE)
%--------------------------------------

%--------------------------------------
% Parameters for the different phases

% Organic phase

mass_polymer_o                  = 8.8; % [g] PLGA
mass_ethylacetate_o             = 44.76; % [g] Ethylacetate
mass_api_o                      = 6.13; % [g] API
mass_benzylalcohol_o            = 18.9; % [g] Benzylalcohol
mass_o                          =
mass_polymer_o+mass_ethylacetate_o+mass_api_o+mass_benzylalcohol_o; % [g]
Mass of the organic phase

% Liquid phase

volume_pva_l                    = 635; % [ml] Polyvinylalcohol [ONLY ON THE
OUTSIDE]
mass_pva_l                      = volume_pva_l/density_pva; %
Polyvinylalcohol
mass_ethylacetate_l             = 45; % [g] Ethylacetate
mass_l                          = mass_pva_l+mass_ethylacetate_l; % [g]
Mass of the liquid phase

% Quench solution

volume_water_q                  = 3440; % [ml] Water
mass_water_q                    = volume_water_q/density_water_5; % [g]
Water
mass_ethylacetate_q             = 83.03; % [g] Ethylacetate
%--------------------------------------

%--------------------------------------
% Dimensions and operating parameter of the experimental vessel

n_revolutions                   = 1000 % [1/min] Revolutions per minute
d_stirrer                       = 112*10^(-3); % [m] Diameter of the
stirrer
d_particle                      = 2*r_max; % [m] Diameter of the sphere
```

```
v_relative                      =
pi*(n_revolutions/60)*d_stirrer*(d_particle/d_stirrer)^(1/3); % [m/s]
Relative velocity

% Reynolds, Schmidt and Sherwood number for the ethylacetate

Re                              =
(v_relative*d_particle)/Viscosity_water_kin; % Reynolds number
Sc_EA                           = Viscosity_water_kin/D_EA_WC; % Schmidt
number
Sh_EA                           = 2 + 0.347*(Re)^(0.62)*(Sc_EA)^(0.31); %
Sherwood number by Steinberger/Treybal et al.

% Reynolds, Schmidt and Sherwood number for the benzylalcohol

Sc_BA                           = Viscosity_water_kin/D_BA_WC; % Schmidt
number
Sh_BA                           = 2 + 0.347*(Re)^(0.62)*(Sc_BA)^(0.31); %
Sherwood number by Steinberger/Treybal et al.

% Mass transfer coefficent

betafl_EA                       = (Sh_EA*D_EA_WC)/d_particle; % [m/s] Mass
transfer coefficient of ethylacetate
betafl_BA                       = (Sh_BA*D_BA_WC)/d_particle; % [m/s]
Masstransport coefficent of benzylalcohol
betafl_W                        = 1 * 10^(-2); % [m/s] Masstransport
coefficent of water
Surface_particle                = radius(1,r_intervall)^2*4*pi; % [m^2]
Initial surface of the sphere
%--------------------------------------

%--------------------------------------
% Parameters for the shrinking function

mass_disp_initial               = (mass_o); % [g] Initial mass of the whole
disperse phase
volume_particle_intial          = r_max^3*pi*4/3; % [m^3] Volume of the
sphere at the beginning (t=0)
mass_particle_initial           = density_disp*volume_particle_intial; %
[kg] Mass of one sphere at the beginning (t=0)
volume_disp_inital              = mass_disp_initial*10^-3/density_disp; %
[m^3] Initial volume of the whole disperse phase

number_of_particles_m           = mass_disp_initial*10^-
3/mass_particle_initial; % Number of spheres
number_of_particles_v           =
volume_disp_inital/volume_particle_intial; % Number of spheres

mass_ethylacetate_q_temp        = 0;
mass_benzylalcohol_q_temp       = 0;
%--------------------------------------

%--------------------------------------
% Volumefraction in the quench solution

ambient_volume                  = ((mass_water_q/density_water_5 +
((mass_ethylacetate_q + mass_ethylacetate_l) / density_ea_5) +
mass_pva_l/density_pva));
```

```
ambientconc_EA                = ((mass_ethylacetate_q +
mass_ethylacetate_l) / density_ea_5)/ambient_volume; % [m^3 EA/m^3 Quench
Solution] Volumefraction of Ethylacetate in the quench solution
ambientconc_BA                =  0; % [m^3 BA/m^3 Quench Solution]
Volumefraction of Benzylalcohol in the quench solution
ambientconc_W                 = (mass_water_q /
density_water_5)/ambient_volume; % [m^3 W/m^3 Quench Solution]
Volumefraction of Water in the quench solution
ambientconc_PVA               = (mass_pva_l/density_pva)/ambient_volume;
% [m^3 PVA/m^3 Quench Solution] Volumefraction of PVA in the quench
solution

Sum_ambient                   =
ambientconc_BA+ambientconc_EA+ambientconc_PVA+ambientconc_W % Sum ambient
concentrations must be 1
%--------------------------------------

%--------------------------------------
% PDEPE parameter

m                             = 2; % PDEPE parameter
%--------------------------------------

%--------------------------------------
% Initial volumefraction and massfraction in the sphere at the beginning
(t=0)

volume_f_initial_EA           = ((mass_ethylacetate_o * 10^-3) /
density_ea_5)/volume_disp_inital; % [m^3 EA/m^3 DP]Volumefraction of
ethylacetate in the sphere at the beginning (t=0)
mass_f_initial_EA             = mass_ethylacetate_o /mass_o; % [kg EA/kg
DP] Massfraction of ethylacetate in the sphere at the beginning (t=0)

volume_f_initial_BA           = ((mass_benzylalcohol_o * 10^-3) /
density_ba)/volume_disp_inital; % [m^3 BA/m^3 DP] Volumefraction of
benzylalcohol in the sphere at the beginning (t=0)
mass_f_initial_BA             = mass_benzylalcohol_o / mass_o; % [kg
BA/kg DP] Massfraction of benzylalcohol in the sphere at the beginning
(t=0)

volume_f_initial_PLGA         = ((mass_polymer_o * 10^-3) /
density_polymer)/volume_disp_inital; % [m^3 Polymer/m^3 DP]
Volumefraction of Polymer (PLGA)
mass_f_initial_PLGA           = mass_polymer_o / mass_o; % [kg Polymer/kg
DP] Massfraction of Polymer (PLGA)

volume_f_initial_W            = 0; % [m^3 W/m^3 DP]Volumefraction of
water in the sphere at the beginning (t=0)
mass_f_initial_W              = 0; % [kg W/kg DP] Massfraction of water
in the sphere at the beginning (t=0)

mass_f_initial_API            = mass_api_o / mass_o; % [kg API/kg DP]
Massfraction of API
volume_f_initial_API          = 1 - (volume_f_initial_EA +
volume_f_initial_BA + volume_f_initial_PLGA + volume_f_initial_W); % [m^3
API/m^3 DP] Volumefraction of API


% API
```

```
density_api                     = (density_disp -
density_ea_5*volume_f_initial_EA - density_ba*volume_f_initial_BA -
density_polymer*volume_f_initial_PLGA)/volume_f_initial_API % [g/cm^3]
Density of the api (NOT REAL VALUE)

%display(density_api);

if density_api < 0

    display('Density Polymer wrong');
    stop;

end

volume_f_initial_API_c          = volume_f_initial_API -
mass_f_initial_API*density_disp/density_api;

% Density of the disperse phase without the ethylaceate and benzylalcohol
volume_f_po_api                 = volume_f_initial_API +
volume_f_initial_PLGA;

volume_f_API                    = volume_f_initial_API/volume_f_po_api;
volume_f_PLGA                   = volume_f_initial_PLGA/volume_f_po_api;

sum_volume_f                    = volume_f_API + volume_f_PLGA;

if sum_volume_f ~= 1

    display('Volume fraction wrong !!');
    stop;

end

density_po_api                  = volume_f_API * density_api +
volume_f_PLGA * density_polymer;  % [g/cm^3] Density of the
benzylalcohol, api and polymer solution

%display(density_po_api );

density_disp_controll           =
volume_f_initial_API*density_api+volume_f_initial_BA*density_ba+volume_f_
initial_EA*density_ea_5+volume_f_initial_PLGA*density_polymer; % [g/cm^3]
Controll the calculated values

if density_disp_controll ~= density_disp

    display('Mass fraction wrong !!');
    stop;
end

% Sum of all volumefractions at the beginning must be 1
sum_mass_f_initial              = mass_f_initial_EA + mass_f_initial_BA +
mass_f_initial_PLGA + mass_f_initial_API + mass_f_initial_W;
sum_volume_f_initial            = volume_f_initial_EA + volume_f_initial_BA
+ volume_f_initial_API + volume_f_initial_PLGA + volume_f_initial_W;

if sum_mass_f_initial ~= 1
```

```
    display('Mass fraction wrong !!');
    stop;
end

if sum_volume_f_initial ~= 1

    display('Volume fraction wrong !!');
    stop;

end

% Mass distribution of the shells at the beginning t=0

for n=1:r_intervall-1

     volume_areas(1,n)                = 4/3*pi*(radius(1,(n+1))^3-
     radius(1,n)^3); % [m^3] Volume of the spherical shells

     mass_volume_areas_init(1,n)      =
     volume_areas(1,n).*density_disp_vector(1,n); % [kg] Mass of every
     spherical cell at the beginning

end

% Controll weight function

Sum_mass                          = sum(mass_volume_areas_init); % [kg]
Sum the mass of the shells at the beginning t=0

Sum_mass_controll                 = (mass_o*10^-3)/number_of_particles_m;
% [kg] Define the initial mass of every sphere

Error_sum_mass                    = Sum_mass/Sum_mass_controll; % Error of
the calculation

%display(Error_sum_mass);

delta_m_EA_temp                   = 0;
delta_m_BA_temp                   = 0;
%---------------------------------------

%---------------------------------------
% Shrinking parameters

radialPositions               = radius; % [m] Define the radial positions
volume_f_current_EA           =
linspace(volume_f_initial_EA,volume_f_initial_EA,r_intervall); % Initial
volume fraction current vector for ethylacetate
volume_f_current_BA           =
linspace(volume_f_initial_BA,volume_f_initial_BA,r_intervall); % Initial
volume fraction current vector for benzylalcohol
%---------------------------------------

%---------------------------------------
% Define Memory matrixes and variables

memVolumefraction_EA_plot     = zeros(r_intervall,t_intervall+1);
```

```
memVolumefraction_BA_plot     = zeros(r_intervall,t_intervall+1);
memRadius_plot                = zeros(r_intervall,t_intervall+1);
memMass_plot                  = zeros(1,t_intervall+1);
memTime_plot                  = zeros(1,t_intervall+1);
memAmbient_EA_plot            = zeros(1,t_intervall+1);
memAmbient_BA_plot            = zeros(1,t_intervall+1);
membetafl_EA_plot             = zeros(1,t_intervall+1);
membetafl_BA_plot             = zeros(1,t_intervall+1);
memDiffusion_plot             = zeros(1,t_intervall+1);

memVolumefraction_EA_plot(:,1)= volume_f_current_EA'; % Define the first
row of the new volumefraction matrix
memVolumefraction_BA_plot(:,1)= volume_f_current_BA'; % Define the first
row of the new volumefraction matrix
memRadius_plot(:,1)           = radius'; % [m] Define the first row of
the radius matrix
memMass_plot(1,1)             = r_max^3*4/3*pi*density_disp; % [kg]
Define the mass of the sphere at the beginning (t=0)
memTime_plot(1,t_intervall+1) = t_total_max; % [s] Define the start time
memAmbient_EA_plot(1,1)       = ambientconc_EA; % Define the
volumefraction distribution of the quench solution at the beginning (t=0)
memAmbient_BA_plot(1,1)       = ambientconc_BA; % Define the
volumefraction distribution of the quench solution at the beginning (t=0)
membetafl_EA_plot(1,1)        = betafl_EA; % [m/s] Define the mass
transport coefficient at the beginning (t=0)
membetafl_BA_plot(1,1)        = betafl_BA; % [m/s] Define the mass
transport coefficient at the beginning (t=0)
memDiffusion_plot(1,1)        = D_EA_WC; % [m^2/s] Define the diffusion
coefficient at the beginning (t=0)
%--------------------------------------

%--------------------------------------
% Diffusioncoefficents of the different components and there constants

diffusion_coefficent_EA       = D_EA_WC; % [m^2/s] Diffusion coefficient
of ethylacetate
diffusion_c_EA_begin          = -7;
diffusion_c_EA_end            = -15;
r_EA_1                        = 1.7496; % Constant depending on the
system
r_EA_3                        = -((diffusion_c_EA_begin)-
(diffusion_c_EA_end))/(volume_f_initial_PLGA-1) % Constant depending on
the system
r_EA_2                        = -(diffusion_c_EA_end)-r_EA_3 % Constant
depending on the system



diffusion_coefficent_BA       = D_BA_WC; % [m^2/s] Diffusion coefficient
of benzylalcohol
diffusion_c_BA_begin          = -7;
diffusion_c_BA_end            = -15;
r_BA_1                        = 1.6916; % Constant depending on the
system
r_BA_3                        = -((diffusion_c_BA_begin)-
(diffusion_c_BA_end))/(volume_f_initial_PLGA-1) % Constant depending on
the system
r_BA_2                        = -(diffusion_c_BA_end)-r_BA_3 % Constant
depending on the system
```

```
diffusion_coefficent_W        = 10^-10;  % [m^2/s] Diffusion coefficient
of water
%--------------------------------------

%--------------------------------------
% CFL_Number

velocity = (betafl_EA*density_ea_5*volume_f_initial_EA +
betafl_BA*density_ba*volume_f_initial_BA)/density_disp;
CFL = velocity*delta_t/(d_measured/(r_intervall*2));

display(CFL);
%--------------------------------------
```

## DiffusionCoefficient.m

```
% Diffusion Coefficent
% Empirical correlation from HAYDUK and MINHAS and WILKE and Chang
% (c) by Hannes Pucher
% Sources : Perry's handbook of chemical engineering, VDI, infosticker on
% the bottle of the chemicals
%--------------------------------------


% Reaction Parameters
temp                = 278.15; % Temperatur of the solutions [K]

% Parameters of Ethylacetate
M_ethyl             = 88.1; % Molar mass of ethylacetate [g/mol]
T_melt_ethyl        = -82.4; % Melting point of ethylacetate [°C]
T_boiling_ethyl     = 77.1; % Boiling Point of ethylacetate [°C]
density_ethyl       = 0.901; % Density of ethylacetate [g/cm^3]
molar_volume_ethyl  = M_ethyl/density_ethyl; % Molar Volume of
ethylacetate [cm^3/mol]

% Parameters of Benzylalcohol
M_benzyl            = 108.14; % Molar mass of benzylalcohol [g/mol]
T_melt_benzyl       = -15.3; % Melting point of benzylalcohol [°C]
T_boiling_benzyl    = 204.7; % Boiling Point of benzylalcohol [°C]
density_benzyl      = 1.0455; % Density of benzylalcohol [g/cm^3]
molar_volume_benzyl = M_benzyl/density_benzyl; % Molar Volume of
benzylalcohol [cm^3/mol]

% Parameters of Water
M_water             = 18.0153; % Molar mass of water [g/mol]
T_melt_water        = 0.01; % Melting point of water [°C]
T_boiling_water     = 99.974; % Boiling Point of water [°C]
Viscosity_water     = 1518.1 * 10^-6; % [Pa*s] or [kg/(m*s)] VDI 5 Dba 4;
Dynamic viscosity of water at 5 [°C]
Viscosity_water_calc= Viscosity_water*10^3; % Dynamic viscosity of water
[cp]
Viscosity_water_kin = 1.518 * 10^(-6); % [m^2/s] VDI 5 Dba 4; Kinematic
Viscosity of water at 5 [°C]
ass_factor          = 2.26; % Association factor of solvent water that
accounts for hydrogen bonding

% Correlation for the diffusion coefficent by WILKE - CHANG, Perry's
% handbook of chemical engineering 5-51

D_EA_WC             = (7.4*10^(-8)*(ass_factor*M_water/1000)^(1/2)*temp)/
(Viscosity_water_calc*(molar_volume_ethyl)^(0.6))

D_BA_WC             = (7.4*10^(-8)*(ass_factor*M_water/1000)^(1/2)*temp)/
(Viscosity_water_calc*(molar_volume_benzyl)^(0.6))
```

```
% Correlation for the diffusion coefficent by HAYDUK and MINHAS;
% Multicomponent Mass Transfer 1993 by Ross Taylor, R. Krishna

%epsilon_b_EA        = 9.58/molar_volume_ethyl-1.12;
%D_EA_HM             = 1.25*10^(-8)*(molar_volume_ethyl^(-0.19)-
0.292)*temp^(1.52)*Viscosity_water_calc^(epsilon_b_EA) % Dissusion
coefficent of ethylacetate [m^2/s]

%epsilon_b_BA        = 9.58/molar_volume_benzyl-1.12;
%D_BA_HM             = 1.25*10^(-8)*(molar_volume_benzyl^(-0.19)-
0.292)*temp^(1.52)*Viscosity_water_calc^(epsilon_b_BA) % Diffusion
coefficent of benzylalcohol [m^2/s]
```

## Calcpde.m

```
% Numerical Solution Calcpde
% (c) by Hannes Pucher
%--------------------------------------

% Definition of the c,f,s parameter for the PDEPE Solver

function [c,f,s, diffusion_coefficent_EA] = calcpde(radius, t,
volume_fraction, Dvolume_fractionDradius)

    % Definition of the global parameters

    global diffusion_coefficent_BA
    global diffusion_coefficent_W
    global r_EA_1
    global r_EA_2
    global r_EA_3
    global r_BA_1
    global r_BA_2
    global r_BA_3
    global volume_f_initial_API

    % Diffusion coefficients
    diffusion_coefficent_EA =  r_EA_1.*10.^(-(r_EA_2+r_EA_3*(1-
    volume_fraction(1,1)-volume_fraction(2,1)-volume_f_initial_API)));%
    [m^2/s] Diffusion Coefficient of ethylacetate
    diffusion_coefficent_BA =  r_BA_1.*10.^(-(r_BA_2+r_BA_3*(1-
    volume_fraction(1,1)-volume_fraction(2,1)-volume_f_initial_API)));%
    [m^2/s] Diffusion Coefficient of benzylalcohol

    c = [1;1;1];
    f = [diffusion_coefficent_EA; diffusion_coefficent_BA;
    diffusion_coefficent_W].*Dvolume_fractionDradius;
    s = [0;0;0];

end
```

## Calcic.m

```
% Numerical Solution Initial Condition for the PDEPE
% (c) by Hannes Pucher
%--------------------------------------

function volume_fraction_0 = calcic(radius)

     % Definition of the global parameters

     global volume_f_initial_EA
     global volume_f_initial_BA
     global volume_f_initial_W
     global radialPositions
     global volume_f_current_EA
     global volume_f_current_BA

     % Define the initial condition to solve the PDE

     volume_f_initial_EA = interp1(radialPositions, volume_f_current_EA,
     radius, 'spline');
     volume_f_initial_BA = interp1(radialPositions, volume_f_current_BA,
     radius, 'spline');
     volume_f_initial_W  = interp1(radialPositions, volume_f_current_EA,
     radius, 'spline');

     volume_fraction_0   = [volume_f_initial_EA; volume_f_initial_BA;
     volume_f_initial_W];

end
```

## Calcbc.m

```
% Numerical Solution Boundary Condition for the PDEPE
% (c) by Hannes Pucher
%--------------------------------------

function [pl,ql,pr,qr] = calcbc(radius_l, volume_fraction_l, radius_r,
volume_fraction_r,t)

     % Definition of the global parameters

     global betafl_EA
     global betafl_BA
     global betafl_W
     global ambientconc_W
     global Surface_particle
     global number_of_particles_m

     % Center of the microparticle (r=0)

     pl=[0;0;0];
     ql=[1;1;1];

     % Surface of the microparticle (r=r_max)

     %Surface_particle = 1;
     ambientconc_EA_bc = 0;
     ambientconc_BA_bc = 0;
     %number_of_particles_m=1;

     pr=[betafl_EA*(volume_fraction_r(1)-
     ambientconc_EA_bc);betafl_BA*(volume_fraction_r(2)-
     ambientconc_BA_bc);betafl_W*(volume_fraction_r(3)-ambientconc_W)];%-
     equilibrium_conc;
     qr=[1;1;1];

end
```

## Shrinking.m

```
% Numerical Model for Shrinking
% (c) by Hannes Pucher
%--------------------------------------

%--------------------------------------
% Define the different matrixes used in the calculation

volume_fraction_area_EA                 = zeros(t_span,r_intervall-1);
volume_fraction_area_BA                 = zeros(t_span,r_intervall-1);
volume_areas                            = zeros(1,r_intervall-1);
volume_distribution                     = zeros(1,r_intervall-1);
volume_matrix_EA                        = zeros(t_span-1,r_intervall-1);
volume_matrix_BA                        = zeros(t_span-1,r_intervall-1);
Weigth_new                              = zeros(1,t_span-1);
radius_new_cell                         = zeros(1,t_span);
radius_new_cell_temp                    = zeros(1,t_span-1);
delta_radius                            = zeros(1,t_span-1);
mass_volume_areas                       = zeros(1,r_intervall-1);
%--------------------------------------

%--------------------------------------
%  Define the volume and mass difference of the ethylacetate from one
%  timestep to another

for n=1:r_intervall-1

    for u=1:t_span-1

     volume_fraction_area_EA(:,n)     = (volume_fraction_EA(:,n) +
     volume_fraction_EA(:,n+1))./2; % Mean volumefraction of two adjacent
     cells

     volume_areas(1,n)                = 4/3*pi*(radius(1,(n+1))^3-
     radius(1,n)^3); % [m^3] Volume of the spherical shells

     mass_volume_areas(1,n)           =
     volume_areas(1,n).*density_disp_vector(1,n); % [kg] Mass of every
     spherical cell at the beginning

     volume_matrix_EA(u,n)            = ((volume_fraction_area_EA(u,n) -
     volume_fraction_area_EA(u+1,n))); % Volume fraction difference
     between each timestep

     Sum_Volume_EA                    = sum(volume_matrix_EA); % Sum the
     volume matrix to get the whole difference from one timestep to
     another

    end

end
%--------------------------------------

%--------------------------------------
%  Define the volume and mass difference of the benzylalcohol from one
%  timestep to another
```

```
for n=1:r_intervall-1

    for u=1:t_span-1

     volume_fraction_area_BA(:,n)      = (volume_fraction_BA(:,n) +
     volume_fraction_BA(:,n+1))./2; % Mean volumefraction of two adjacent
     cells

     volume_matrix_BA(u,n)             = ((volume_fraction_area_BA(u,n) -
     volume_fraction_area_BA(u+1,n))); % Volume fraction difference
     between each timestep

     Sum_Volume_BA                     = sum(volume_matrix_BA); % Sum the
     volume matrix to get the whole difference from one timestep to
     another

    end

end
%--------------------------------------

%--------------------------------------
% Define the mass and volume loss in every shell and define the new
radius
% after delta t

%sum_volumefraction                       = volume_f_current_EA

density_new                              = density_ea_5 *
volume_fraction_area_EA(t_span,:) + density_ba *
volume_fraction_area_BA(t_span,:) + density_po_api*(1-
volume_fraction_area_EA(t_span,:)-volume_fraction_area_BA(t_span,:));
%[kg/m^3] New density of the spherical cells

mass_fraction_initial_EA                 =
(density_ea_5*volume_fraction_area_EA(1,:))./density_disp_vector(1,:); %
Inital mass fraction distribution of ethylacetate

mass_fraction_final_EA                   =
(density_ea_5*volume_fraction_area_EA(t_span,:))./density_new; % Final
mass fraction distribution of ethylacetate

delta_mass_fraction_EA                   = mass_fraction_initial_EA-
mass_fraction_final_EA; % Difference of the mass fraction of ehtylacetate

delta_m_EA                               =
mass_volume_areas_init.*delta_mass_fraction_EA; % [kg] Mass difference of
ethylacetate in every spherical shell

sum_mass_EA                              = sum(delta_m_EA); % [kg] Mass
loss of ethylacetate of one sphere

sum_delta_m_EA                           = delta_m_EA + delta_m_EA_temp;


mass_fraction_initial_BA                 =
(density_ba*volume_fraction_area_BA(1,:))./density_disp_vector(1,:); %
Inital mass fraction distribution of benzylalcohol
```

```
mass_fraction_final_BA                      =
(density_ba*volume_fraction_area_BA(t_span,:))./density_new; % Final mass
fraction distribution of benzylalcohol

delta_mass_fraction_BA                      = mass_fraction_initial_BA-
mass_fraction_final_BA; % Difference of the mass fraction of
benzylalcohol

delta_m_BA                                  =
mass_volume_areas_init.*delta_mass_fraction_BA; % [kg] Mass difference of
benzylalcohol in every spherical shell

sum_mass_BA                                 = sum(delta_m_BA); % [kg] Mass
loss of benzylalcohol of one sphere

sum_delta_m_BA                              = delta_m_BA + delta_m_BA_temp;


mass_volume_areas_new                       = mass_volume_areas_init -
sum_delta_m_EA  - sum_delta_m_BA ; % [kg] New mass of every sperical
shell

sum_mass                                    = sum(mass_volume_areas_new); %
[kg] Mass of the new sphere

Volume_areas_new                            =
mass_volume_areas_new./density_disp_controll; % [m^3] Define the new
volume of every spherical cell

delta_volume                                = volume_areas -
Volume_areas_new; % [m^3] Delta volume


[temp dim_delta_volume] = size(delta_volume);

sum_Sum_volume = zeros(size(delta_volume));

for j = 1:dim_delta_volume

    for k = 1:j

     sum_Sum_volume(j) = sum_Sum_volume(j) + delta_volume(k); % [m^3] Sum
     delta volume matrix to get the whole difference from one timestep to
     another

    end

end

clear temp;


for x = 1:t_span-1

     radius_new_cell_temp(1,x)      = ((-3/(4*pi)*sum_Sum_volume(1,x)) +
     radius(1,x+1)^3)^(1/3); % [m] New Radius of every shell
```

```
    radius_new_cell(1,1)            = radius(1,1); % [m] Define the first
    element of the new radial vector
    radius_new_cell(1,x+1)          = radius_new_cell_temp(1,x); % [m]
    Define the new radius of ever sphere cell
end
%--------------------------------------

%--------------------------------------
% Define the new parameters needed for further calculations

radius_new                          = radius_new_cell(1,t_span); %
[m] Radius of the new sphere
Surface_particle_new                = radius_new^2*pi*4; % [m^2]
Surface of the new sphere
density_disp_vector                 = density_new; % [kg/m^3] Density
of the new sphere

% Define the mass loss and the new ambient concentration

mass_volume_areas                   = mass_volume_areas_new; % [kg]
Mass of every spherical sphere
mass_ethylacetate_q_new             = mass_ethylacetate_q_temp +
sum_mass_EA*number_of_particles_m * 10^3; % [g] Mass loss of ethylacetate
mass_benzylalcohol_q_new            = mass_benzylalcohol_q_temp +
sum_mass_BA*number_of_particles_m * 10^3; % [g] Mass loss of
benzylalcohol

% Controll the mass balance

mass_ethylacetate_sphere            =
sum(mass_fraction_final_EA.*mass_volume_areas_init)*number_of_particles_m
*10^3; % [g] Ethylacetate in the sphere at the end
mass_benzylalcohol_sphere           =
sum(mass_fraction_final_BA.*mass_volume_areas_init)*number_of_particles_m
*10^3; % [g] Benzylalcohol in the sphere at the end

error_ethylacetate                  = (mass_ethylacetate_o-
mass_ethylacetate_q_new-
mass_ethylacetate_sphere)/mass_ethylacetate_o*100; % Error of the mass
balance of EA must be 0
error_benzylalcohol                 = (mass_benzylalcohol_o-
mass_benzylalcohol_q_new-
mass_benzylalcohol_sphere)/mass_benzylalcohol_o*100; % Error of the mass
balance of BA  must be 0

%density_quench                      = %[kg/m^3] New density of the
quench solution

ambientconc_EA                      = ((mass_ethylacetate_l +
mass_ethylacetate_q + mass_ethylacetate_q_new) /
density_ea_5)/((mass_water_q + mass_ethylacetate_q +
mass_ethylacetate_q_new + mass_l + mass_benzylalcohol_q_new) /
density_quench); % [m^3 EA/m^3 Quench Solution] New ambient concentration
ambientconc_BA                      = ((mass_benzylalcohol_q_new) /
density_ba)/((mass_water_q + mass_ethylacetate_q +
mass_ethylacetate_q_new + mass_benzylalcohol_q_new + mass_l) /
density_quench); % [m^3 BA/m^3 Quench Solution] New ambient concentration

% Dimensions and operating parameter of the experimental vessel
```

```
d_particle              = 2*radius_new; % [m] Diameter of the sphere
v_relative              =
pi*(n_revolutions/60)*d_stirrer*(d_particle/d_stirrer)^(1/3); % [m/s]
Relative velocity
Re                      = (v_relative*d_particle)/Viscosity_water_kin; %
Reynolds number

% Reynolds, Schmidt and Sherwood number for the ethylacetate

Sc_EA                   = Viscosity_water_kin/D_EA_WC; % Schmidt number
Sh_EA                   = 2 + 0.347*(Re)^(0.62)*(Sc_EA)^(0.31); %
Sherwood number by Steinberger/Treybal et al.

% Reynolds, Schmidt and Sherwood number for the benzylalcohol

Sc_BA                   = Viscosity_water_kin/D_BA_WC; % Schmidt number
Sh_BA                   = 2 + 0.347*(Re)^(0.62)*(Sc_BA)^(0.31); %
Sherwood number by Steinberger/Treybal et al.

% Mass transfer coefficent

betafl_EA               = (Sh_EA*D_EA_WC)/d_particle; % [m/s] Mass
transfer coefficient of ethylacetate
betafl_BA               = (Sh_BA*D_BA_WC)/d_particle; % [m/s]
Masstransport coefficent of benzylalcohol
%--------------------------------------
```

# II. Plot Files

## Plot_Radius.m

```
% Plot file for Radius for the Numerical Solution
% (c) by Hannes Pucher
%-------------------------------------


% Creation of the different matrixes used in the calculation
[temp dim_memTime]                  = size(memTime); % Get the size of the
memTime vector
memTime_temp                        = zeros(1,dim_memTime+1);
memRadius_temp                      = zeros(1,dim_memTime+1);

% Resize and rearange the time and radius vector

for p = 1:dim_memTime

     memTime_temp(1,1)                = t(1,1); % [s] Starting time
     memTime_temp(1,p+1)              = memTime(1,p); % [s] Rearange the
     time vector

     memRadius_temp(1,1)              = r_max; % [m] Radius at the
     beginning
     memRadius_temp(1,p+1)            = memRadius_end(1,p); % [m] Rearange
     the radius vector

end

memTime                             = memTime_temp; % Define the variables
used in the plot

memRadius_end                       = memRadius_temp; % Define the
variables used in the plot

for o = 1:t_intervall

     memVolumefraction_EA_plot(:,o+1) =
     memVolumefraction_EA(:,o*value_get); % Volume matrix fraction of the
     ethylacetate

     memVolumefraction_BA_plot(:,o+1) =
     memVolumefraction_BA(:,o*value_get); % Volume matrix fraction of the
     benzylalcohol

     volume_fraction_rest              = 1 - memVolumefraction_EA_plot -
     memVolumefraction_BA_plot; % Volume fraction of the rest
     (Polymere,API,BA,Polyvinylalcohol)

     memRadius_plot(:,o+1)             = memRadius(:,o*value_get); %
     [m]Radius matrix of the new sphere

     memMass_plot(1,o+1)               = memMass(1,o*value_get); % [kg]
     Mass vector of the new sphere
```

```
    memTime_plot(1,o)                = memTime(1,(o-1)*value_get+1); %
    [s] Rearangend time vector

    memAmbient_EA_plot(1,o+1)        = memAmbient_EA(1,o*value_get); %
    [m^3 EA/m^3 Quench Solution] New volumefraction vector of
    ethylacetate in the quench solution

    memAmbient_BA_plot(1,o+1)        = memAmbient_BA(1,o*value_get); %
    [m^3 BA/m^3 Quench Solution] New volumefraction vector of
    benzylalcohol in the quench solution

    membetafl_EA_plot(1,o+1)         = membetafl_EA(1,o*value_get); %
    [m/s] Mass transport coefficient of ethylacetate

    membetafl_BA_plot(1,o+1)         = membetafl_BA(1,o*value_get); %
    [m/s] Mass transport coefficient of benzylalcohol

end
%-------------------------------------

%-------------------------------------
% Defines the normalized radius
for q = 1:t_intervall+1

    memRadius_plot_normalized(:,q)=
    memRadius_plot(:,q)/memRadius_plot(t_span,1); % Normalized radius
    matrix

end
%-------------------------------------

%-------------------------------------
% Data from 12/16.08.2010
QICPIC_data_1      = [163.49 115 111.675 105.03 99.5 105 96.25 99.985
92.596 100.165].*10^-6; % [m] SMD measured with QICPIC
QICPIC_time_1      = [0 12 50 85 145 205 265 270 315 1170]; % [min]
QICPIC time
QICPIC_time_1_plot = QICPIC_time_1 * 60; % [sec] QICPIC time

HELOS_data_1       = [163.49 116 96 110 96.66 104 106.01 107].*10^-6; %
[m] SMD measured with HELOS
HELOS_time_1       = [0 50 85 225 265 270 315 1170]; % [sec] HELOS time
HELOS_time_1_plot  = HELOS_time_1 * 60; % [sec] HELOS time


% Data from 19.11.2010
HELOS_data_2       = [163.49 135.36 121.29 118.43 123.99 123.55 128.15
106.68 114.01 118.61].*10^-6; % [m] SMD measured with HELOS
HELOS_time_2       = [0 3.66 9.66 30.5 45.33 60.08 72.76 90.08 300 1200];
% [min] HELOS time
HELOS_time_2_plot  = HELOS_time_2 * 60; % [sec] HELOS time
%-------------------------------------

%-------------------------------------
% Save all variables
save All_Parameters.mat
%-------------------------------------
```

```
%--------------------------------------
% Plot Radius vs Time
figure(1)
plot(QICPIC_time_1_plot/3600,QICPIC_data_1*10^6,'bd','MarkerSize',17,'Mar
kerFaceColor','b')
hold on
plot(HELOS_time_1_plot/3600,HELOS_data_1*10^6,'gs','MarkerSize',17,'Marke
rFaceColor','g')
%hold on
%plot(HELOS_time_2_plot/3600,HELOS_data_2*10^6,'ro','MarkerSize',17,'Mark
erFaceColor','r')
hold on
plot(memTime/3600,memRadius_end*2*10^6,'k','LineWidth',2)

% Label and Title of the plot
title('', 'FontSize',35)
xlabel('Time [h]', 'FontSize',20)
ylabel('SMD [\mum]', 'FontSize',20)
set(gca, 'FontSize', 20)

% Legend
legend('QICPIC','HELOS','Location','NorthEast')

% Define the intervall of the axis
axis([0 t_total_max/3600 0*10^2 1.8*10^2])

% Resize the plot to fullscreen
set(gcf,'position',get(0,'screensize'))

% Save Figure as SMD.tiff
saveas(gcf,'SMD.tiff', 'tiffn');
%--------------------------------------

%--------------------------------------
% Plot Radius vs Volumefraction of the Polymere
figure(2)
plot(memRadius_plot_normalized,volume_fraction_rest,'LineWidth',2)

% Label and Title of the plot
title('', 'FontSize',35)
% Real Radial Positions
%xlabel('Radius [m]', 'FontSize',30)

% Normalized Radial Positions
xlabel('Normalized Radius', 'FontSize',20)

ylabel('\phi_{Remaining Components}', 'FontSize',20)
set(gca, 'FontSize', 20)

% Legend
legend('\tau = 0','\tau = 0.1','\tau = 0.2','\tau = 0.3','\tau =
0.4','\tau = 0.5','\tau = 0.6','\tau = 0.7','\tau = 0.8','\tau =
0.9','\tau = 1.0','Location','EastOutside')

% Resize the plot to fullscreen
set(gcf,'position',get(0,'screensize'))

% Save Figure as Volumefraction_Poly.tiff
saveas(gcf,'Volumefraction_Poly.tiff', 'tiffn');
```

```
%----------------------------------
```

## Plot_Diffusion_EA.m

```
% Plot file for Diffusion Coefficient of Ethylacetate
% (c) by Hannes Pucher
%--------------------------------------


% Definition of the Diffusion Coefficient Matrix
diffusion_coefficent_EA=zeros(r_intervall,t_intervall+1);

for u=1:r_intervall

    for n=1:t_intervall+1

     diffusion_coefficent_EA(u,n) = r_EA_1.*10.^(-
     (r_EA_2+r_EA_3*(volume_fraction_rest(u,n)-volume_f_initial_API)));

    end

end

% Plot Radius vs Diffusion Coefficient

figure(11)
semilogy(memRadius_plot_normalized,diffusion_coefficent_EA,'LineWidth',2)
;

% Label and Title of the plot
title('', 'FontSize',30)

% Real Radial Positions
%xlabel('Radius [m]', 'FontSize',30)

% Normalized Radial Positions
xlabel('Normalized Radius', 'FontSize',20)
ylabel('D_{EA} [m^2/s]', 'FontSize',20)
set(gca, 'FontSize', 20)

% Legend
legend('\tau = 0','\tau = 0.1','\tau = 0.2','\tau = 0.3','\tau =
0.4','\tau = 0.5','\tau = 0.6','\tau = 0.7','\tau = 0.8','\tau =
0.9','\tau = 1.0','Location','EastOutside')

% Resize the plot to fullscreen
set(gcf,'position',get(0,'screensize'))

% Save Figure as diffusion_coefficent_EA.tiff
saveas(gcf,'diffusion_coefficent_EA.tiff', 'tiffn');

hold on
```

## Plot_Diffusion_BA.m

```
% Plot file for Diffusion Coefficient of Benzy alcohol
% (c) by Hannes Pucher
%--------------------------------------


% Definition of the Diffusion Coefficient Matrix
diffusion_coefficent_BA=zeros(r_intervall,t_intervall+1);

for u=1:r_intervall

    for n=1:t_intervall+1

     diffusion_coefficent_BA(u,n) = r_BA_1.*10.^(-
     (r_BA_2+r_BA_3*(volume_fraction_rest(u,n)-volume_f_initial_API)));

    end

end

% Plot Radius vs Diffusion Coefficient

figure(12)
semilogy(memRadius_plot_normalized,diffusion_coefficent_BA,'LineWidth',2)
;

% Label and Title of the plot
title('', 'FontSize',30)

% Real Radial Positions
%xlabel('Radius [m]', 'FontSize',30)

% Normalized Radial Positions
xlabel('Normalized Radius', 'FontSize',20)
ylabel('D_{BA} [m^2/s]', 'FontSize',20)
set(gca, 'FontSize', 20)

% Legend
legend('\tau = 0','\tau = 0.1','\tau = 0.2','\tau = 0.3','\tau =
0.4','\tau = 0.5','\tau = 0.6','\tau = 0.7','\tau = 0.8','\tau =
0.9','\tau = 1.0','Location','EastOutside')

% Resize the plot to fullscreen
set(gcf,'position',get(0,'screensize'))

% Save Figure as diffusion_coefficent_BA.tiff
saveas(gcf,'diffusion_coefficent_BA.tiff', 'tiffn');

hold on
```

## Plot_EA.m

```
% Plot file for Ethylacetate for the Numerical Solution
% (c) by Hannes Pucher
%--------------------------------------

% Ethylacetate

%--------------------------------------
% Plot matrixs for the ambient concentration and mass transport
coefficient
memAmbient_EA_plot_fine = zeros(1,t_total+1);
memAmbient_EA_plot_fine(1,1) = memAmbient_EA_plot(1,1);

membetafl_EA_plot_fine = zeros(1,t_total+1);
membetafl_EA_plot_fine(1,1) = membetafl_EA_plot(1,1);

for z = 2:t_total+1

     memAmbient_EA_plot_fine(1,z) = memAmbient_EA(1,z-1);
     membetafl_EA_plot_fine(1,z) = membetafl_EA(1,z-1);

end
%--------------------------------------

%--------------------------------------
% Plot Volume fraction in the quench solution vs time

figure(31)
plot(memTime/3600,memAmbient_EA_plot_fine,'LineWidth',2)

% Label and Title of the plot
title('', 'FontSize',35)
xlabel('Time [h]', 'FontSize',20)
ylabel('\phi_{Ethylacetate in Continuous Phase}', 'FontSize',20)
set(gca, 'FontSize', 20)

% Define the intervall of the axis
axis([0 t_total/3600 0 6*10^-2])

% Resize the plot to fullscreen
set(gcf,'position',get(0,'screensize'))

% Save Figure as Volumefraction_EA_in_the_Quench.tiff
saveas(gcf,'Volumefraction_EA_in_the_Quench.tiff', 'tiffn');
%--------------------------------------

%--------------------------------------
% Plot Radius vs Volume fraction

figure(32)
plot(memRadius_plot_normalized,memVolumefraction_EA_plot,'LineWidth',2)

% Label and Title of the plot
title('', 'FontSize',35)

% Normalized Radial Positions
```

```
xlabel('Normalized Radius', 'FontSize',20)

ylabel('\phi_{Ethylacetate}', 'FontSize',20)
set(gca, 'FontSize', 20)
set(gca,'YTickLabel',{'0','0.10','0.20','0.30','0.40','0.50','0.60','0.70
',})

% Legend
legend('{\tau} = 0','{\tau} = 0.1','{\tau} = 0.2','{\tau} = 0.3','{\tau}
= 0.4','{\tau} = 0.5','{\tau} = 0.6','{\tau} = 0.7','{\tau} =
0.8','{\tau} = 0.9','{\tau} = 1.0','Location','EastOutside')

% Define the intervall of the axis
%axis([0 1 0.0 0.8])

% Resize the plot to fullscreen
set(gcf,'position',get(0,'screensize'))

% Save Figure as Volumefraction_EA.tiff
saveas(gcf,'Volumefraction_EA.tiff', 'tiffn');
%--------------------------------------

%--------------------------------------
% Plot Time vs Delta Mass

figure(33)
plot(memTime_plot/3600,memMass_plot,'LineWidth',2)

% Label and Title of the plot
title('', 'FontSize',35)
xlabel('Time [h]', 'FontSize',20)
ylabel('m_{S} [kg]', 'FontSize',20)
set(gca, 'FontSize', 20)

% Define the intervall of the axis
%axis([0 5 0 3*10^-12])

% Resize the plot to fullscreen
set(gcf,'position',get(0,'screensize'))

% Save Figure as Mass_Sphere.tiff
saveas(gcf,'Mass_Sphere.tiff', 'tiffn');
%--------------------------------------

%--------------------------------------
% Plot Time vs Mass Transport Coefficient

figure(34)
plot(memTime/3600,membetafl_EA_plot_fine,'LineWidth',2)

% Label and Title of the plot
title('', 'FontSize',35)
xlabel('Time [h]', 'FontSize',20)
ylabel('\beta_{Ethylacetate} [m/s]', 'FontSize',20)
set(gca, 'FontSize', 20)
%set(gca,'YTickLabel',{'0','0.10','0.20','0.30','0.40','0.50','0.60','0.7
0',})

% Define the intervall of the axis
```

```
axis([0 t_total/3600 5*10^-3 10*10^-3])

% Resize the plot to fullscreen
set(gcf,'position',get(0,'screensize'))

% Save Figure as MassTransportCoefficient_EA.tiff
saveas(gcf,'MassTransportCoefficient_EA.tiff', 'tiffn');
%--------------------------------------
```

## Plot_BA.m

```
% Plot file for Benzylalcohol for the Numerical Solution
% (c) by Hannes Pucher
%--------------------------------------

% Benzylalcohol

%--------------------------------------
% Plot matrixs for the ambient concentration and mass transport
coefficient
memAmbient_BA_plot_fine = zeros(1,t_total+1);
memAmbient_BA_plot_fine(1,1) = memAmbient_BA_plot(1,1);

membetafl_BA_plot_fine = zeros(1,t_total+1);
membetafl_BA_plot_fine(1,1) = membetafl_BA_plot(1,1);

for z = 2:t_total+1

     memAmbient_BA_plot_fine(1,z) = memAmbient_BA(1,z-1);
     membetafl_BA_plot_fine(1,z) = membetafl_BA(1,z-1);

end
%--------------------------------------

%--------------------------------------
% Plot Volume fraction in the quench solution vs time

figure(21)
plot(memTime/3600,memAmbient_BA_plot_fine,'LineWidth',2)

% Label and Title of the plot
title('', 'FontSize',35)
xlabel('Time [h]', 'FontSize',20)
ylabel('\phi_{Benzylalcohol in Continuous Phase}', 'FontSize',20)
set(gca, 'FontSize', 20)

% Define the intervall of the axis
axis([0 t_total/3600 0 0.005])

% Resize the plot to fullscreen
set(gcf,'position',get(0,'screensize'))

% Save Figure as Volumefraction_BA_in_the_Quench.tiff
saveas(gcf,'Volumefraction_BA_in_the_Quench.tiff', 'tiffn');
%--------------------------------------

%--------------------------------------
% Plot Radius vs Volume fraction

figure(22)
plot(memRadius_plot_normalized,memVolumefraction_BA_plot,'LineWidth',2)

% Label and Title of the plot
title('', 'FontSize',35)
```

```
% Normalized Radial Positions
xlabel('Normalized Radius', 'FontSize',20)

ylabel('\phi_{Benzylalcohol}', 'FontSize',20)
set(gca, 'FontSize', 20)
set(gca,'YTickLabel',{'0','0.05','0.10','0.15','0.20','0.25'})

% Legend
legend('{\tau} = 0','{\tau} = 0.1','{\tau} = 0.2','{\tau} = 0.3','{\tau}
= 0.4','{\tau} = 0.5','{\tau} = 0.6','{\tau} = 0.7','{\tau} =
0.8','{\tau} = 0.9','{\tau} = 1.0','Location','EastOutside')

% Define the intervall of the axis
%axis([0 1 0 0.3])

% Resize the plot to fullscreen
set(gcf,'position',get(0,'screensize'))

% Save Figure as Volumefraction_BA.tiff
saveas(gcf,'Volumefraction_BA.tiff', 'tiffn');
%--------------------------------------

%--------------------------------------
% Plot Time vs Mass Transport Coefficient

figure(23)
plot(memTime/3600,membetafl_BA_plot_fine,'LineWidth',2)

% Label and Title of the plot
title('', 'FontSize',35)
xlabel('Time [h]', 'FontSize',20)
ylabel('\beta_{Benzylalcohol} [m/s]', 'FontSize',20)
set(gca, 'FontSize', 20)

% Define the intervall of the axis
axis([0 t_total/3600 5*10^-3 10*10^-3])

% Resize the plot to fullscreen
set(gcf,'position',get(0,'screensize'))

% Save Figure as MassTransportCoefficient_BA.tiff
saveas(gcf,'MassTransportCoefficient_BA.tiff', 'tiffn');
%--------------------------------------
```

## Surf_Plot_EA.m

```
% Surf plot file for EA
% (c) by Hannes Pucher
%--------------------------------------

%--------------------------------------
% Define the new time matirx for the 3D Plot

mem_time_surf = zeros(t_span,11);

for n = 1 : t_span

     mem_time_surf(n,:)=memTime_plot/3600';

end
%--------------------------------------

%--------------------------------------
% Surf Plot Time vs Radius vs Volume fraction
figure(200)
surf(mem_time_surf,memRadius_plot*10^6,memVolumefraction_EA_plot)

% Label and Title of the plot
title('', 'FontSize',35)
xlabel('Time [h]', 'FontSize',20)
ylabel('Radius', 'FontSize',20)
zlabel('\phi_{Ethyl acetate}', 'FontSize',20)
set(gca, 'FontSize', 20)

view([125 15]);

% Legend
%legend('QICPIC','HELOS 1','HELOS 2','Location','NorthEast')

% Define the intervall of the axis
%axis([0 t_total_max 0 85])

% Resize the plot to fullscreen
set(gcf,'position',get(0,'screensize'))

% Save Figure as 3DPlot.tiff
saveas(gcf,'3DPlot_EA.tiff', 'tiffn');
%--------------------------------------
```

## Surf_Plot_BA.m

```
% Surf plot file for BA
% (c) by Hannes Pucher
%--------------------------------------

%--------------------------------------
% Define the new time matirx for the 3D Plot

mem_time_surf = zeros(t_span,11);

for n = 1 : t_span

     mem_time_surf(n,:)=memTime_plot/3600';

end
%--------------------------------------

%--------------------------------------
% Surf Plot Time vs Radius vs Volume fraction
figure(200)
surf(mem_time_surf,memRadius_plot*10^6,memVolumefraction_BA_plot)

% Label and Title of the plot
title('', 'FontSize',35)
xlabel('Time [h]', 'FontSize',20)
ylabel('Radius', 'FontSize',20)
zlabel('\phi_{Benzyl alcohol}', 'FontSize',20)
set(gca, 'FontSize', 20)

view([125 15]);

% Legend
%legend('QICPIC','HELOS 1','HELOS 2','Location','NorthEast')

% Define the intervall of the axis
%axis([0 t_total_max 0 85])

% Resize the plot to fullscreen
%set(gcf,'position',get(0,'screensize'))

% Save Figure as 3DPlot.tiff
saveas(gcf,'3DPlot_BA.tiff', 'tiffn');
%--------------------------------------
```

## Surf_Plot_RC.m

```
% Surf plot file for RC
% (c) by Hannes Pucher
%--------------------------------------

%--------------------------------------
% Define the new time matirx for the 3D Plot

mem_time_surf = zeros(t_span,11);

for n = 1 : t_span

    mem_time_surf(n,:)=memTime_plot/3600';

end
%--------------------------------------

%--------------------------------------
% Surf Plot Time vs Radius vs Volume fraction
figure(200)
surf(mem_time_surf,memRadius_plot*10^6,volume_fraction_rest)

% Label and Title of the plot
title('', 'FontSize',35)
xlabel('Time [h]', 'FontSize',20)
ylabel('Radius', 'FontSize',20)
zlabel('\phi_{Remaining Components}', 'FontSize',20)
set(gca, 'FontSize', 20)

view([310 25]);

% Legend
%legend('QICPIC','HELOS 1','HELOS 2','Location','NorthEast')

% Define the intervall of the axis
%axis([0 t_total_max 0 85])

% Resize the plot to fullscreen
set(gcf,'position',get(0,'screensize'))

% Save Figure as 3DPlot.tiff
saveas(gcf,'3DPlot_RC.tiff', 'tiffn');
%--------------------------------------
```

# III. Parameter Studies

## Sensivity_Study.m

```
% Sensivity Study for the Time Step Size and the Number of Radial
Sections
% (c) by Hannes Pucher
%--------------------------------------

%--------------------------------------
% Clear, Close and delete all
clc
clear all
close all
%--------------------------------------

%--------------------------------------
% Load memTime_plot files in the workspace
load('memTime_plot_tspan=10')
memTime_plot_tspan_10 = memTime_plot;
load('memTime_plot_tspan=25')
memTime_plot_tspan_25 = memTime_plot;
load('memTime_plot_tspan=50')
memTime_plot_tspan_50 = memTime_plot;


% Load files for t_span=10
load('memRadius_plot_1_tspan=10.mat')
memRadius_plot_1_tspan_10 = memRadius_plot;
load('memRadius_plot_0.5_tspan=10.mat')
memRadius_plot_0_5_tspan_10 = memRadius_plot;
load('memRadius_plot_0.1_tspan=10.mat')
memRadius_plot_0_1_tspan_10 = memRadius_plot;
load('memRadius_plot_0.05_tspan=10.mat')
memRadius_plot_0_05_tspan_10 = memRadius_plot;

load('Volumefraction_Poly_1_tspan=10.mat')
Volume_fraction_Polymere_1_tspan_10 = volume_fraction_Rest;
load('Volumefraction_Poly_0.5_tspan=10.mat')
Volume_fraction_Polymere_0_5_tspan_10 = volume_fraction_Rest;
load('Volumefraction_Poly_0.1_tspan=10.mat')
Volume_fraction_Polymere_0_1_tspan_10 = volume_fraction_Rest;
load('Volumefraction_Poly_0.05_tspan=10.mat')
Volume_fraction_Polymere_0_05_tspan_10 = volume_fraction_Rest;


% Load files for t_span = 25
load('memRadius_plot_1_tspan=25.mat')
memRadius_plot_1_tspan_25 = memRadius_plot;
load('memRadius_plot_0.5_tspan=25.mat')
memRadius_plot_0_5_tspan_25 = memRadius_plot;
load('memRadius_plot_0.1_tspan=25.mat')
memRadius_plot_0_1_tspan_25 = memRadius_plot;
load('memRadius_plot_0.05_tspan=25.mat')
memRadius_plot_0_05_tspan_25 = memRadius_plot;

load('Volumefraction_Poly_1_tspan=25.mat')
```

```
Volume_fraction_Polymere_1_tspan_25 = volume_fraction_Rest;
load('Volumefraction_Poly_0.5_tspan=25.mat')
Volume_fraction_Polymere_0_5_tspan_25 = volume_fraction_Rest;
load('Volumefraction_Poly_0.1_tspan=25.mat')
Volume_fraction_Polymere_0_1_tspan_25 = volume_fraction_Rest;
load('Volumefraction_Poly_0.05_tspan=25.mat')
Volume_fraction_Polymere_0_05_tspan_25 = volume_fraction_Rest;


% Load files for t_span = 50
load('memRadius_plot_1_tspan=50.mat')
memRadius_plot_1_tspan_50 = memRadius_plot;
load('memRadius_plot_0.5_tspan=50.mat')
memRadius_plot_0_5_tspan_50 = memRadius_plot;
load('memRadius_plot_0.1_tspan=50.mat')
memRadius_plot_0_1_tspan_50 = memRadius_plot;
load('memRadius_plot_0.05_tspan=50.mat')
memRadius_plot_0_05_tspan_50 = memRadius_plot;

load('Volumefraction_Poly_1_tspan=50.mat')
Volume_fraction_Polymere_1_tspan_50 = volume_fraction_Rest;
load('Volumefraction_Poly_0.5_tspan=50.mat')
Volume_fraction_Polymere_0_5_tspan_50 = volume_fraction_Rest;
load('Volumefraction_Poly_0.1_tspan=50.mat')
Volume_fraction_Polymere_0_1_tspan_50 = volume_fraction_Rest;
load('Volumefraction_Poly_0.05_tspan=50.mat')
Volume_fraction_Polymere_0_05_tspan_50 = volume_fraction_Rest;


% Load files for t_span = 100
load('memRadius_plot_1_tspan=100.mat')
memRadius_plot_1_tspan_100 = memRadius_plot;
load('memRadius_plot_0.5_tspan=100.mat')
memRadius_plot_0_5_tspan_100 = memRadius_plot;
load('memRadius_plot_0.1_tspan=100.mat')
memRadius_plot_0_1_tspan_100 = memRadius_plot;
load('memRadius_plot_0.05_tspan=100.mat')
memRadius_plot_0_05_tspan_100 = memRadius_plot;

load('Volumefraction_Poly_1_tspan=100.mat')
Volume_fraction_Polymere_1_tspan_100 = volume_fraction_Rest;
load('Volumefraction_Poly_0.5_tspan=100.mat')
Volume_fraction_Polymere_0_5_tspan_100 = volume_fraction_Rest;
load('Volumefraction_Poly_0.1_tspan=100.mat')
Volume_fraction_Polymere_0_1_tspan_100 = volume_fraction_Rest;
load('Volumefraction_Poly_0.05_tspan=100.mat')
Volume_fraction_Polymere_0_05_tspan_100 = volume_fraction_Rest;

% Load files for t_span = 200
load('memRadius_plot_1_tspan=200.mat')
memRadius_plot_1_tspan_200 = memRadius_plot;
load('memRadius_plot_0.5_tspan=200.mat')
memRadius_plot_0_5_tspan_200 = memRadius_plot;
load('memRadius_plot_0.1_tspan=200.mat')
memRadius_plot_0_1_tspan_200 = memRadius_plot;
load('memRadius_plot_0.05_tspan=200.mat')
memRadius_plot_0_05_tspan_200 = memRadius_plot;

load('Volumefraction_Poly_1_tspan=200.mat')
Volume_fraction_Polymere_1_tspan_200 = volume_fraction_Rest;
```

```
load('Volumefraction_Poly_0.5_tspan=200.mat')
Volume_fraction_Polymere_0_5_tspan_200 = volume_fraction_Rest;
load('Volumefraction_Poly_0.1_tspan=200.mat')
Volume_fraction_Polymere_0_1_tspan_200 = volume_fraction_Rest;
load('Volumefraction_Poly_0.05_tspan=200.mat')
Volume_fraction_Polymere_0_05_tspan_200 = volume_fraction_Rest;
%--------------------------------------

%--------------------------------------
% Define the different t_spans and matrixes for the plots
t_span_10 = 10;
t_span_25 = 25;
t_span_50 = 50;
t_span_100 = 100;
t_span_200 = 200;

radius_end_dt_0_05 =
[memRadius_plot_0_05_tspan_10(t_span_10,11)*2,memRadius_plot_0_05_tspan_2
5(t_span_25,11)*2,memRadius_plot_0_05_tspan_50(t_span_50,11)*2,memRadius_
plot_0_05_tspan_100(t_span_100,11)*2,memRadius_plot_0_05_tspan_200(t_span
_200,11)*2]*10^6;
radius_end_dt_0_1 =
[memRadius_plot_0_1_tspan_10(t_span_10,11)*2,memRadius_plot_0_1_tspan_25(
t_span_25,11)*2,memRadius_plot_0_1_tspan_50(t_span_50,11)*2,memRadius_plo
t_0_1_tspan_100(t_span_100,11)*2,memRadius_plot_0_1_tspan_200(t_span_200,
11)*2]*10^6;
radius_end_dt_0_5 =
[memRadius_plot_0_5_tspan_10(t_span_10,11)*2,memRadius_plot_0_5_tspan_25(
t_span_25,11)*2,memRadius_plot_0_5_tspan_50(t_span_50,11)*2,memRadius_plo
t_0_5_tspan_100(t_span_100,11)*2,memRadius_plot_0_5_tspan_200(t_span_200,
11)*2]*10^6;
radius_end_dt_1 =
[memRadius_plot_1_tspan_10(t_span_10,11)*2,memRadius_plot_1_tspan_25(t_sp
an_25,11)*2,memRadius_plot_1_tspan_50(t_span_50,11)*2,memRadius_plot_1_ts
pan_100(t_span_100,11)*2,memRadius_plot_1_tspan_200(t_span_200,11)*2]*10^
6;

radius_end_t_span_10 =
[memRadius_plot_0_05_tspan_10(t_span_10,11)*2,memRadius_plot_0_1_tspan_10
(t_span_10,11)*2,memRadius_plot_0_5_tspan_10(t_span_10,11)*2,memRadius_pl
ot_1_tspan_10(t_span_10,11)*2]*10^6;
radius_end_t_span_25 =
[memRadius_plot_0_05_tspan_25(t_span_25,11)*2,memRadius_plot_0_1_tspan_25
(t_span_25,11)*2,memRadius_plot_0_5_tspan_25(t_span_25,11)*2,memRadius_pl
ot_1_tspan_25(t_span_25,11)*2]*10^6;
radius_end_t_span_50 =
[memRadius_plot_0_05_tspan_50(t_span_50,11)*2,memRadius_plot_0_1_tspan_50
(t_span_50,11)*2,memRadius_plot_0_5_tspan_50(t_span_50,11)*2,memRadius_pl
ot_1_tspan_50(t_span_50,11)*2]*10^6;
radius_end_t_span_100 =
[memRadius_plot_0_05_tspan_100(t_span_100,11)*2,memRadius_plot_0_1_tspan_
100(t_span_100,11)*2,memRadius_plot_0_5_tspan_100(t_span_100,11)*2,memRad
ius_plot_1_tspan_100(t_span_100,11)*2]*10^6;
radius_end_t_span_200 =
[memRadius_plot_0_05_tspan_200(t_span_200,11)*2,memRadius_plot_0_1_tspan_
200(t_span_200,11)*2,memRadius_plot_0_5_tspan_200(t_span_200,11)*2,memRad
ius_plot_1_tspan_200(t_span_200,11)*2]*10^6;

radius_10s_t_span_10 =
[memRadius_plot_0_05_tspan_10(t_span_10,2)*2,memRadius_plot_0_1_tspan_10(
```

```
t_span_10,2)*2,memRadius_plot_0_5_tspan_10(t_span_10,2)*2,memRadius_plot_
1_tspan_10(t_span_10,2)*2]*10^6;
radius_10s_t_span_25 =
[memRadius_plot_0_05_tspan_25(t_span_25,2)*2,memRadius_plot_0_1_tspan_25(
t_span_25,2)*2,memRadius_plot_0_5_tspan_25(t_span_25,2)*2,memRadius_plot_
1_tspan_25(t_span_25,2)*2]*10^6;
radius_10s_t_span_50 =
[memRadius_plot_0_05_tspan_50(t_span_50,2)*2,memRadius_plot_0_1_tspan_50(
t_span_50,2)*2,memRadius_plot_0_5_tspan_50(t_span_50,2)*2,memRadius_plot_
1_tspan_50(t_span_50,2)*2]*10^6;
radius_10s_t_span_100 =
[memRadius_plot_0_05_tspan_100(t_span_100,2)*2,memRadius_plot_0_1_tspan_1
00(t_span_100,2)*2,memRadius_plot_0_5_tspan_100(t_span_100,2)*2,memRadius
_plot_1_tspan_100(t_span_100,2)*2]*10^6;
radius_10s_t_span_200 =
[memRadius_plot_0_05_tspan_200(t_span_200,2)*2,memRadius_plot_0_1_tspan_2
00(t_span_200,2)*2,memRadius_plot_0_5_tspan_200(t_span_200,2)*2,memRadius
_plot_1_tspan_200(t_span_200,2)*2]*10^6;

% Define r_intervall and delta_t
r_intervall = [t_span_10,t_span_25,t_span_50,t_span_100,t_span_200]*2;
delta_t = [0.05,0.1,0.5,1];
%--------------------------------------

%--------------------------------------
% Plot the particle radius over the number of radial sections for the
% different delta t
figure(90)
plot(r_intervall,radius_end_dt_0_05,'-
ks','LineWidth',1,'MarkerSize',17,'MarkerFaceColor','k')
hold on
plot(r_intervall,radius_end_dt_0_1,'-
gs','LineWidth',1,'MarkerSize',12,'MarkerFaceColor','g')
hold on
plot(r_intervall,radius_end_dt_0_5,'-
rs','LineWidth',1,'MarkerSize',7,'MarkerFaceColor','r')
hold on
plot(r_intervall,radius_end_dt_1,'-
bs','LineWidth',1,'MarkerSize',2,'MarkerFaceColor','b')

% Label and Title of the plot
title('', 'FontSize',35)
xlabel('Number of radial Sections', 'FontSize',20)
ylabel('SMD_{t = 50 sec} [\mum]', 'FontSize',20)
set(gca, 'FontSize', 20)
%set(gca,'YTickLabel',{'0','0.10','0.20','0.30','0.40','0.50','0.60','0.7
0',})

% Legend
legend('\Delta t = 0.05 [sec]','\Delta t = 0.1 [sec]','\Delta t = 0.5
[sec]','\Delta t = 1.0 [sec]','Location','NorthEast')

% Define the intervall of the axis
axis([0 410 130 150])

% Resize the plot to fullscreen
set(gcf,'position',get(0,'screensize'))

% Save Figure as r_int_radius.tiff
saveas(gcf,'radius_trend.tiff', 'tiffn');
```

```
%--------------------------------------

%--------------------------------------
% Plot the particle radius over the delta t for the different numbers of
% radial sections
figure(100)
plot(delta_t,radius_end_t_span_10,'-
ks','LineWidth',1,'MarkerSize',17,'MarkerFaceColor','k')
hold on
plot(delta_t,radius_end_t_span_25,'-
gs','LineWidth',1,'MarkerSize',17,'MarkerFaceColor','g')
hold on
plot(delta_t,radius_end_t_span_50,'-
rs','LineWidth',1,'MarkerSize',17,'MarkerFaceColor','r')
hold on
plot(delta_t,radius_end_t_span_100,'-
bs','LineWidth',1,'MarkerSize',17,'MarkerFaceColor','b')
hold on
plot(delta_t,radius_end_t_span_200,'-
ys','LineWidth',1,'MarkerSize',17,'MarkerFaceColor','y')

% Label and Title of the plot
title('', 'FontSize',35)
xlabel('\Delta t', 'FontSize',20)
ylabel('SMD_{t = 50 sec} [\mum]', 'FontSize',20)
set(gca, 'FontSize', 20)

% Legend
legend('Radial Sections = 20','Radial Sections = 50','Radial Sections =
100','Radial Sections = 200','Radial Sections =
400','Location','NorthEast')

% Define the intervall of the axis
axis([0.0 1.1 130 160])

% Resize the plot to fullscreen
set(gcf,'position',get(0,'screensize'))

% Save Figure as dt_radius.tiff
saveas(gcf,'dt_radius.tiff', 'tiffn');
%--------------------------------------

%--------------------------------------
% Plot the particle radius after 10 seconds over the delta t for the
% different numbers of radial sections
figure(110)
plot(delta_t,radius_10s_t_span_10,'-
ks','LineWidth',1,'MarkerSize',17,'MarkerFaceColor','k')
hold on
plot(delta_t,radius_10s_t_span_25,'-
gs','LineWidth',1,'MarkerSize',17,'MarkerFaceColor','g')
hold on
plot(delta_t,radius_10s_t_span_50,'-
rs','LineWidth',1,'MarkerSize',17,'MarkerFaceColor','r')
hold on
plot(delta_t,radius_10s_t_span_100,'-
bs','LineWidth',1,'MarkerSize',17,'MarkerFaceColor','b')
hold on
plot(delta_t,radius_10s_t_span_200,'-
ys','LineWidth',1,'MarkerSize',17,'MarkerFaceColor','y')
```

```
% Label and Title of the plot
title('', 'FontSize',35)
xlabel('\Delta t', 'FontSize',20)
ylabel('SMD_{t = 10 sec} [\mum]', 'FontSize',20)
set(gca, 'FontSize', 20)

% Legend
legend('Radial Sections = 20','Radial Sections = 50','Radial Sections =
100','Radial Sections = 200','Radial Sections =
400','Location','NorthEast')

% Define the intervall of the axis
axis([0.0 1.1 140 170])

% Resize the plot to fullscreen
set(gcf,'position',get(0,'screensize'))

% Save Figure as dt_radius_10s.tiff
saveas(gcf,'dt_radius_10s.tiff', 'tiffn');
%--------------------------------------
```

## CFL_Analysis.m

```
% CFL-Analysis
% (c) by Hannes Pucher
%--------------------------------------

%--------------------------------------
% Clear, Close and delete all
clc
clear all
close all
%--------------------------------------

%--------------------------------------
% Load files in the workspace
load('memTime_plot.mat')

% Load files for the different CFL Numbers and 260 rpm
load('memRadius_CFL_3_24.mat')
memRadius_plot_CFL_3_24 = memRadius_plot*10^6;
load('memRadius_CFL_6_48.mat')
memRadius_plot_CFL_6_48 = memRadius_plot*10^6;
load('memRadius_CFL_647_7.mat')
memRadius_plot_CFL_647_7 = memRadius_plot*10^6;
load('memRadius_CFL_6477.mat')
memRadius_plot_CFL_6477 = memRadius_plot*10^6;

load('memvolumefraction _CFL_3_24.mat')
Volumefraction_Poly_CFL_3_24 = volume_fraction_rest;
load('memvolumefraction _CFL_6_48.mat')
Volumefraction_Poly_CFL_6_48 = volume_fraction_rest;
load('memvolumefraction _CFL_647_7.mat')
Volumefraction_Poly_CFL_647_7 = volume_fraction_rest;
load('memvolumefraction _CFL_6477.mat')
Volumefraction_Poly_CFL_6477 = volume_fraction_rest;
%--------------------------------------

%--------------------------------------
% Define t_span, CFL vector and SMD_beginning
t_span_100 = 100;

SMD_beginning = 163.49*10^-6; % [m] SMD of the micro-particle at the
beginning t=0
CFL_vector = [3.24;6.48;347.7;6477];
Time_vector = [33.37;15.73;0.66;0.07];
%--------------------------------------

%--------------------------------------
% Define the SMD after 1 second for different CFL numbers

memRadius_CFL_3_24 = memRadius_plot_CFL_3_24(t_span_100,11)*2;
memRadius_plot_CFL_3_24_end =
memRadius_plot_CFL_3_24(t_span_100,11)*2/memRadius_CFL_3_24;
memRadius_plot_CFL_6_48_end =
memRadius_plot_CFL_6_48(t_span_100,11)*2/memRadius_CFL_3_24;
memRadius_plot_CFL_647_7_end =
memRadius_plot_CFL_647_7(t_span_100,11)*2/memRadius_CFL_3_24;
```

```
memRadius_plot_CFL_6477_end =
memRadius_plot_CFL_6477(t_span_100,2)*2/memRadius_CFL_3_24;

memDeviation_vector =
[memRadius_plot_CFL_3_24_end;memRadius_plot_CFL_6_48_end;memRadius_plot_C
FL_647_7_end;memRadius_plot_CFL_6477_end]-1;
memDeviation_vector = memDeviation_vector*100;
%--------------------------------------

%--------------------------------------
% Plot the different SMD for the different CFL numbers over time
figure(90)
plot(memTime_plot,memRadius_plot_CFL_3_24(t_span_100,:)*2,'g',memTime_plo
t,memRadius_plot_CFL_6_48(t_span_100,:)*2,'b:',memTime_plot,memRadius_plo
t_CFL_647_7(t_span_100,:)*2,'r--','LineWidth',3)

% Label and Title of the plot
%title('w=0', 'FontSize',25)
xlabel('Time [s]', 'FontSize',20)
ylabel('SMD [\mum]', 'FontSize',20)
set(gca, 'FontSize', 20)

% Legend
legend('CFL = 3.24','CFL = 6.48','CFL = 647.7','Location','East')

% Define the intervall of the axis
axis([0 1 155 165])

% Resize the plot to fullscreen
set(gcf,'position',get(0,'screensize'))

% Save Figure as w=0.tiff
saveas(gcf,'w=260.tiff', 'tiffn');
%--------------------------------------

%--------------------------------------
% Compare the final micro-particle radius after 1 second for different
CFL
% numbers
figure(100)
semilogx(CFL_vector,memDeviation_vector,'-
rs','LineWidth',1,'MarkerSize',17,'MarkerFaceColor','r')

% Label and Title of the plot
%title('w=0', 'FontSize',25)
xlabel('CFL Number', 'FontSize',20)
ylabel('Deviation [%]', 'FontSize',20)
set(gca, 'FontSize', 20)

% Legend
%legend('CFL = 6.48','CFL = 3.24','CFL = 647.7','w = 50','w = 100','w =
250','w = 500','Location','East')

% Define the intervall of the axis
%axis([0 50 0 200*10^-3])

% Resize the plot to fullscreen
set(gcf,'position',get(0,'screensize'))
```

```
% Save Figure as w=0.tiff
saveas(gcf,'CFL_compare_SMD_Deviation_1sec.tiff', 'tiffn');
%--------------------------------------

%--------------------------------------
% Compare the simulation time between the different cases
figure(110)
semilogx(CFL_vector,Time_vector,'-
bs','LineWidth',1,'MarkerSize',17,'MarkerFaceColor','b')

% Label and Title of the plot
%title('w=0', 'FontSize',25)
xlabel('CFL Number', 'FontSize',20)
ylabel('Calculation Period  [min]', 'FontSize',20)
set(gca, 'FontSize', 20)

% Legend
%legend('CFL = 6.48','CFL = 3.24','CFL = 647.7','w = 50','w = 100','w =
250','w = 500','Location','East')

% Define the intervall of the axis
%axis([0 50 0 200*10^-3])

% Resize the plot to fullscreen
set(gcf,'position',get(0,'screensize'))

% Save Figure as w=0.tiff
saveas(gcf,'calculationperiod .tiff', 'tiffn');
%--------------------------------------
```

## Sensivity_Analysis

```
% Sensivity Analysis
% (c) by Hannes Pucher
%--------------------------------------

%--------------------------------------
% Clear, Close and delete all
clc
clear all
close all
%--------------------------------------

%--------------------------------------
% Load files in the workspace
load('memTime')

% Load memRadius_plot files
load('memRadius_plot_r3_0.mat')
memRadius_plot_r3_0 = memRadius_plot*10^6;
load('memRadius_plot_r3_2.mat')
memRadius_plot_r3_2 = memRadius_plot*10^6;
load('memRadius_plot_r3_4.mat')
memRadius_plot_r3_4 = memRadius_plot*10^6;
load('memRadius_plot_r3_6.mat')
memRadius_plot_r3_6 = memRadius_plot*10^6;
load('memRadius_plot_r3_8.mat')
memRadius_plot_r3_8 = memRadius_plot*10^6;
load('memRadius_plot_r3_10.mat')
memRadius_plot_r3_10 = memRadius_plot*10^6;
load('memRadius_plot_r3_12.mat')
memRadius_plot_r3_12 = memRadius_plot*10^6;

% Load volumefraction_Poly files
load('Volumefraction_Poly_r3_0.mat')
Volumefraction_Poly_r3_0 = volume_fraction_rest;
load('Volumefraction_Poly_r3_2.mat')
Volumefraction_Poly_r3_2 = volume_fraction_rest;
load('Volumefraction_Poly_r3_4.mat')
Volumefraction_Poly_r3_4 = volume_fraction_rest;
load('Volumefraction_Poly_r3_6.mat')
Volumefraction_Poly_r3_6 = volume_fraction_rest;
load('Volumefraction_Poly_r3_8.mat')
Volumefraction_Poly_r3_8 = volume_fraction_rest;
load('Volumefraction_Poly_r3_10.mat')
Volumefraction_Poly_r3_10 = volume_fraction_rest;
load('Volumefraction_Poly_r3_12.mat')
Volumefraction_Poly_r3_12 = volume_fraction_rest;
%--------------------------------------

%--------------------------------------
% Define the t_span and the different vectors and matrixes for the plot
t_span_100 = 100;

memRadius_plot_r3_0_end = memRadius_plot_r3_0(t_span_100,:);
memRadius_plot_r3_2_end = memRadius_plot_r3_2(t_span_100,:);
memRadius_plot_r3_4_end = memRadius_plot_r3_4(t_span_100,:);
memRadius_plot_r3_6_end = memRadius_plot_r3_6(t_span_100,:);
```

```
memRadius_plot_r3_8_end = memRadius_plot_r3_8(t_span_100,:);
memRadius_plot_r3_10_end = memRadius_plot_r3_10(t_span_100,:);
memRadius_plot_r3_12_end = memRadius_plot_r3_12(t_span_100,:);

Volumefraction_Poly_r3_0_second = Volumefraction_Poly_r3_0(:,2);
Volumefraction_Poly_r3_2_second = Volumefraction_Poly_r3_2(:,2);
Volumefraction_Poly_r3_4_second = Volumefraction_Poly_r3_4(:,2);
Volumefraction_Poly_r3_6_second = Volumefraction_Poly_r3_6(:,2);
Volumefraction_Poly_r3_8_second = Volumefraction_Poly_r3_8(:,2);
Volumefraction_Poly_r3_10_second = Volumefraction_Poly_r3_10(:,2);
Volumefraction_Poly_r3_12_second = Volumefraction_Poly_r3_12(:,2);

memRadius_plot_r3_0_second =
memRadius_plot_r3_0(:,2)/memRadius_plot_r3_0(t_span_100,1);
memRadius_plot_r3_2_second =
memRadius_plot_r3_2(:,2)/memRadius_plot_r3_0(t_span_100,1);
memRadius_plot_r3_4_second =
memRadius_plot_r3_4(:,2)/memRadius_plot_r3_0(t_span_100,1);
memRadius_plot_r3_6_second =
memRadius_plot_r3_6(:,2)/memRadius_plot_r3_0(t_span_100,1);
memRadius_plot_r3_8_second =
memRadius_plot_r3_8(:,2)/memRadius_plot_r3_0(t_span_100,1);
memRadius_plot_r3_10_second =
memRadius_plot_r3_10(:,2)/memRadius_plot_r3_0(t_span_100,1);
memRadius_plot_r3_12_second =
memRadius_plot_r3_12(:,2)/memRadius_plot_r3_0(t_span_100,1);
%--------------------------------------

%--------------------------------------
% Plot the particle radius over time for the different r_3 values
figure(90)
plot(memTime,memRadius_plot_r3_0_end*2,'k','LineWidth',3)
hold on
plot(memTime,memRadius_plot_r3_2_end*2,'g--','LineWidth',3)
hold on
plot(memTime,memRadius_plot_r3_4_end*2,'r','LineWidth',3)
hold on
plot(memTime,memRadius_plot_r3_6_end*2,'y--','LineWidth',3)
hold on
plot(memTime,memRadius_plot_r3_8_end*2,'b','LineWidth',3)
hold on
plot(memTime,memRadius_plot_r3_10_end*2,'c--','LineWidth',3)
hold on
plot(memTime,memRadius_plot_r3_12_end*2,'m','LineWidth',3)

% Label and Title of the plot
title('', 'FontSize',35)
xlabel('Time [s]', 'FontSize',20)
ylabel('SMD [\mum]', 'FontSize',20)
set(gca, 'FontSize', 20)

% Legend
legend('r_{3}=0','r_{3}=2','r_{3}=4','r_{3}=6','r_{3}=8','r_{3}=10','r_{3
}=12','Location','EastOutside')

% Define the intervall of the axis
axis([0 10 80 180])

% Resize the plot to fullscreen
set(gcf,'position',get(0,'screensize'))
```

```
% Save Figure as r_3_radius_compare.tiff
saveas(gcf,'r_3_radius_compare.tiff', 'tiffn');
%--------------------------------------

%--------------------------------------
% Plot the volumen fraction for 1 sec over the radius for the different
r_3 values
figure(100)
plot(memRadius_plot_r3_0_second,Volumefraction_Poly_r3_0_second,'k','Line
Width',3)
hold on
plot(memRadius_plot_r3_2_second,Volumefraction_Poly_r3_2_second,'g--
','LineWidth',3)
hold on
plot(memRadius_plot_r3_4_second,Volumefraction_Poly_r3_4_second,'r','Line
Width',3)
hold on
plot(memRadius_plot_r3_6_second,Volumefraction_Poly_r3_6_second,'y--
','LineWidth',3)
hold on
plot(memRadius_plot_r3_8_second,Volumefraction_Poly_r3_8_second,'b','Line
Width',3)
hold on
plot(memRadius_plot_r3_10_second,Volumefraction_Poly_r3_10_second,'c--
','LineWidth',3)
hold on
plot(memRadius_plot_r3_12_second,Volumefraction_Poly_r3_12_second,'m','Li
neWidth',3)

% Label and Title of the plot
title('', 'FontSize',35)
xlabel('Normalized Radius', 'FontSize',20)
ylabel('\phi_{Remaining Components}', 'FontSize',20)
set(gca, 'FontSize', 20)

% Legend
legend('r_{3}=0','r_{3}=2','r_{3}=4','r_{3}=6','r_{3}=8','r_{3}=10','r_{3
}=12','Location','EastOutside')

% Define the intervall of the axis
%axis([0 1 0.80000000*10^-4 1.80000000*10^-4])

% Resize the plot to fullscreen
set(gcf,'position',get(0,'screensize'))

% Save Figure as r_3_volume_compare.tiff
saveas(gcf,'r_3_volume_compare.tiff', 'tiffn');
%--------------------------------------
```

# IV. Experimental Validation

## Experimental_Validation_1h.m

```
% Experimental Validation 1 hour
% (c) by Hannes Pucher
%-------------------------------------

%-------------------------------------
% Clear, Close and delete all
clc
clear all
close all
%-------------------------------------

%-------------------------------------
% Load files in the workspace

% Load memRadius_plot files
load('memRadius_plot_r3_7_end.mat')
memRadius_plot_r3_7 = memRadius_end*10^6;
load('memRadius_plot_r3_9_end.mat')
memRadius_plot_r3_9 = memRadius_end*10^6;
load('memRadius_plot_r3_11_end.mat')
memRadius_plot_r3_11 = memRadius_end*10^6;

%-------------------------------------
% Define the memTime_plot vector for the plot
memTime_plot = linspace(0,3601,3601);
%-------------------------------------

%-------------------------------------
% Data from 12/16.08.2010
QICPIC_data_1      = [163.49 115 111.675 105.03 99.5 105 96.25 99.985
92.596 100.165]; % [?m] SMD measured with QICPIC
QICPIC_time_1      = [0 12 50 85 145 205 265 270 315 1170]; % [min]
QICPIC time
QICPIC_time_1_plot = QICPIC_time_1 * 60; % [sec] QICPIC time

HELOS_data_1       = [163.49 116 96 110 96.66 104 106.01 107]; % [?m] SMD
measured with HELOS
HELOS_time_1       = [0 50 85 225 265 270 315 1170]; % [sec] HELOS time
HELOS_time_1_plot  = HELOS_time_1 * 60; % [sec] HELOS time
%-------------------------------------

%-------------------------------------
% Save all variables
save Experimental_Parameters.mat
%-------------------------------------

%-------------------------------------
% Plot Radius vs Time
figure(1)
plot(QICPIC_time_1_plot/3600,QICPIC_data_1,'bd','MarkerSize',17,'MarkerFa
ceColor','b')
hold on
```

```
plot(HELOS_time_1_plot/3600,HELOS_data_1,'gs','MarkerSize',17,'MarkerFace
Color','g')
hold on
plot(memTime_plot/3600,memRadius_plot_r3_7*2,'k','LineWidth',2)
hold on
plot(memTime_plot/3600,memRadius_plot_r3_9*2,'r--','LineWidth',2)
hold on
plot(memTime_plot/3600,memRadius_plot_r3_11*2,'b:','LineWidth',2)
%hold on
%plot(memTime_plot/3600,memRadius_plot_r3_13*2,'b:','LineWidth',2)

% Label and Title of the plot
title('', 'FontSize',35)
xlabel('Time [h]', 'FontSize',20)
ylabel('SMD [\mum]', 'FontSize',20)
set(gca, 'FontSize', 20)

% Legend
legend('QICPIC','HELOS','r_{2} = 6.25, r_{3} = 6.75','r_{2} = 6.00, r_{3}
= 9.00','r_{2} = 5.75, r_{3} = 11.25','Location','NorthEast')

% Define the intervall of the axis
axis([0 1 80 180])

% Resize the plot to fullscreen
set(gcf,'position',get(0,'screensize'))

% Save Figure as SMD.tiff
saveas(gcf,'SMD_new.tiff', 'tiffn');
%--------------------------------------
```

## Experimental_Validation_5h.m

```
% Experimental Validation 5 hour
% (c) by Hannes Pucher
%--------------------------------------

%--------------------------------------
% Clear, Close and delete all
clc
clear all
close all
%--------------------------------------

%--------------------------------------
% Load files in the workspace

% Load memRadius_plot files
load('memRadius_plot_r3_7_end.mat')
memRadius_plot_r3_7 = memRadius_end*10^6;
load('memRadius_plot_r3_9_end.mat')
memRadius_plot_r3_9 = memRadius_end*10^6;
load('memRadius_plot_r3_11_end.mat')
memRadius_plot_r3_11 = memRadius_end*10^6;

%--------------------------------------
% Define the memTime_plot vector for the plot
memTime_plot = linspace(0,18001,18001);
%--------------------------------------

%--------------------------------------
% Data from 12/16.08.2010
QICPIC_data_1      = [163.49 115 111.675 105.03 99.5 105 96.25 99.985
92.596 100.165]; % [m] SMD measured with QICPIC
QICPIC_time_1      = [0 12 50 85 145 205 265 270 315 1170]; % [min]
QICPIC time
QICPIC_time_1_plot = QICPIC_time_1 * 60; % [sec] QICPIC time

HELOS_data_1       = [163.49 116 96 110 96.66 104 106.01 107]; % [m] SMD
measured with HELOS
HELOS_time_1       = [0 50 85 225 265 270 315 1170]; % [sec] HELOS time
HELOS_time_1_plot  = HELOS_time_1 * 60; % [sec] HELOS time
%--------------------------------------

%--------------------------------------
% Save all variables
save Experimental_Parameters.mat
%--------------------------------------

%--------------------------------------
% Plot Radius vs Time
figure(1)
plot(QICPIC_time_1_plot/3600,QICPIC_data_1,'bd','MarkerSize',17,'MarkerFa
ceColor','b')
hold on
plot(HELOS_time_1_plot/3600,HELOS_data_1,'gs','MarkerSize',17,'MarkerFace
Color','g')
hold on
plot(memTime_plot/3600,memRadius_plot_r3_7*2,'k','LineWidth',2)
```

```
hold on
plot(memTime_plot/3600,memRadius_plot_r3_9*2,'r--','LineWidth',2)
hold on
plot(memTime_plot/3600,memRadius_plot_r3_11*2,'b:','LineWidth',2)

% Label and Title of the plot
title('', 'FontSize',35)
xlabel('Time [h]', 'FontSize',20)
ylabel('SMD [\mum]', 'FontSize',20)
set(gca, 'FontSize', 20)

% Legend
legend('QICPIC','HELOS','r_{2} = 6.25, r_{3} = 6.75','r_{2} = 6.00, r_{3}
= 9.00','r_{2} = 5.75, r_{3} = 11.25','Location','NorthEast')

% Define the intervall of the axis
axis([0 5 80 180])

% Resize the plot to fullscreen
set(gcf,'position',get(0,'screensize'))

% Save Figure as SMD.tiff
saveas(gcf,'SMD_new.tiff', 'tiffn');
%--------------------------------------
```

## Experimental_Validaiton_20h.m

```
% Experimental Validation 20 hour
% (c) by Hannes Pucher
%--------------------------------------

%--------------------------------------
% Clear, Close and delete all
clc
clear all
close all
%--------------------------------------

%--------------------------------------
% Load files in the workspace

% Load memRadius_plot files
load('memRadius_plot_r3_7_end.mat')
memRadius_plot_r3_7 = memRadius_end*10^6;
load('memRadius_plot_r3_9_end.mat')
memRadius_plot_r3_9 = memRadius_end*10^6;
load('memRadius_plot_r3_11_end.mat')
memRadius_plot_r3_11 = memRadius_end*10^6;

%--------------------------------------
% Define the memTime_plot vector for the plot
memTime_plot = linspace(0,72001,72001);
%--------------------------------------

%--------------------------------------
% Data from 12/16.08.2010
QICPIC_data_1      = [163.49 115 111.675 105.03 99.5 105 96.25 99.985
92.596 100.165]; % [m] SMD measured with QICPIC
QICPIC_time_1      = [0 12 50 85 145 205 265 270 315 1170]; % [min]
QICPIC time
QICPIC_time_1_plot = QICPIC_time_1 * 60; % [sec] QICPIC time

HELOS_data_1       = [163.49 116 96 110 96.66 104 106.01 107]; % [m] SMD
measured with HELOS
HELOS_time_1       = [0 50 85 225 265 270 315 1170]; % [sec] HELOS time
HELOS_time_1_plot  = HELOS_time_1 * 60; % [sec] HELOS time
%--------------------------------------

%--------------------------------------
% Save all variables
save Experimental_Parameters.mat
%--------------------------------------

%--------------------------------------
% Plot Radius vs Time
figure(1)
plot(QICPIC_time_1_plot/3600,QICPIC_data_1,'bd','MarkerSize',17,'MarkerFa
ceColor','b')
hold on
plot(HELOS_time_1_plot/3600,HELOS_data_1,'gs','MarkerSize',17,'MarkerFace
Color','g')
hold on
plot(memTime_plot/3600,memRadius_plot_r3_7*2,'k','LineWidth',2)
```

```
hold on
plot(memTime_plot/3600,memRadius_plot_r3_9*2,'r--','LineWidth',2)
hold on
plot(memTime_plot/3600,memRadius_plot_r3_11*2,'b:','LineWidth',2)

% Label and Title of the plot
title('', 'FontSize',35)
xlabel('Time [h]', 'FontSize',20)
ylabel('SMD [\mum]', 'FontSize',20)
set(gca, 'FontSize', 20)

% Legend
legend('QICPIC','HELOS','r_{2} = 6.25, r_{3} = 6.75','r_{2} = 6.00, r_{3}
= 9.00','r_{2} = 5.75, r_{3} = 11.25','Location','NorthEast')

% Define the intervall of the axis
axis([0 20 80 180])

% Resize the plot to fullscreen
set(gcf,'position',get(0,'screensize'))

% Save Figure as SMD.tiff
saveas(gcf,'SMD_new.tiff', 'tiffn');
%--------------------------------------
```

## Volume_fraction_EA_Comparison.m

```
% Volume fraction EA Comparison
% (c) by Hannes Pucher
%-------------------------------------

%-------------------------------------
% Clear, Close and delete all
clc
clear all
close all
%-------------------------------------

%-------------------------------------
% Load files in the workspace

% r3=6.75
% Load memVolumefraction_EA_plot files
load('memVolumefraction_EA_plot_1h_6_75.mat')
memVolumefraction_EA_0h_6_75 = memVolumefraction_EA_plot(:,1);
load('memVolumefraction_EA_plot_1h_6_75.mat')
memVolumefraction_EA_1h_6_75 = memVolumefraction_EA_plot(:,11);
load('memVolumefraction_EA_plot_5h_6_75.mat')
memVolumefraction_EA_5h_6_75 = memVolumefraction_EA_plot(:,11);
load('memVolumefraction_EA_plot_20h_6_75.mat')
memVolumefraction_EA_20h_6_75 = memVolumefraction_EA_plot(:,11);

% Load memRadius_plot_normalized files
load('memRadius_plot_normalized_1h_6_75.mat')
memRadius_plot_normalized_0h_6_75 = memRadius_plot_normalized(:,1);
load('memRadius_plot_normalized_1h_6_75.mat')
memRadius_plot_normalized_1h_6_75 = memRadius_plot_normalized(:,11);
load('memRadius_plot_normalized_5h_6_75.mat')
memRadius_plot_normalized_5h_6_75 = memRadius_plot_normalized(:,11);
load('memRadius_plot_normalized_20h_6_75.mat')
memRadius_plot_normalized_20h_6_75 = memRadius_plot_normalized(:,11);
%-------------------------------------

%-------------------------------------
% r3=9.00
% Load memVolumefraction_EA_plot files
load('memVolumefraction_EA_plot_1h_9_00.mat')
memVolumefraction_EA_0h_9_00 = memVolumefraction_EA_plot(:,1);
load('memVolumefraction_EA_plot_1h_9_00.mat')
memVolumefraction_EA_1h_9_00 = memVolumefraction_EA_plot(:,11);
load('memVolumefraction_EA_plot_5h_9_00.mat')
memVolumefraction_EA_5h_9_00 = memVolumefraction_EA_plot(:,11);
load('memVolumefraction_EA_plot_20h_9_00.mat')
memVolumefraction_EA_20h_9_00 = memVolumefraction_EA_plot(:,11);

% Load memRadius_plot_normalized files
load('memRadius_plot_normalized_1h_9_00.mat')
memRadius_plot_normalized_0h_9_00 = memRadius_plot_normalized(:,1);
load('memRadius_plot_normalized_1h_9_00.mat')
memRadius_plot_normalized_1h_9_00 = memRadius_plot_normalized(:,11);
load('memRadius_plot_normalized_5h_9_00.mat')
memRadius_plot_normalized_5h_9_00 = memRadius_plot_normalized(:,11);
load('memRadius_plot_normalized_20h_9_00.mat')
```

```
memRadius_plot_normalized_20h_9_00 = memRadius_plot_normalized(:,11);
%--------------------------------------

%--------------------------------------
% r3=11.25
% Load memVolumefraction_EA_plot files
load('memVolumefraction_EA_plot_1h_11_25.mat')
memVolumefraction_EA_0h_11_25 = memVolumefraction_EA_plot(:,1);
load('memVolumefraction_EA_plot_1h_11_25.mat')
memVolumefraction_EA_1h_11_25 = memVolumefraction_EA_plot(:,11);
load('memVolumefraction_EA_plot_5h_11_25.mat')
memVolumefraction_EA_5h_11_25 = memVolumefraction_EA_plot(:,11);
load('memVolumefraction_EA_plot_20h_11_25.mat')
memVolumefraction_EA_20h_11_25 = memVolumefraction_EA_plot(:,11);

% Load memRadius_plot_normalized files
load('memRadius_plot_normalized_1h_11_25.mat')
memRadius_plot_normalized_0h_11_25 = memRadius_plot_normalized(:,1);
load('memRadius_plot_normalized_1h_11_25.mat')
memRadius_plot_normalized_1h_11_25 = memRadius_plot_normalized(:,11);
load('memRadius_plot_normalized_5h_11_25.mat')
memRadius_plot_normalized_5h_11_25 = memRadius_plot_normalized(:,11);
load('memRadius_plot_normalized_20h_11_25.mat')
memRadius_plot_normalized_20h_11_25 = memRadius_plot_normalized(:,11);
%--------------------------------------

%--------------------------------------
% Save all variables
save Experimental_Parameters.mat
%--------------------------------------

%--------------------------------------
% Plot Radius vs Volume fraction

figure(32)
plot(memRadius_plot_normalized_0h_6_75,memVolumefraction_EA_0h_6_75','k-
o','LineWidth',1)
hold on
plot(memRadius_plot_normalized_1h_6_75,memVolumefraction_EA_1h_6_75','b-
o','LineWidth',1)
hold on
plot(memRadius_plot_normalized_5h_6_75,memVolumefraction_EA_5h_6_75','r-
o','LineWidth',1)
hold on
plot(memRadius_plot_normalized_20h_6_75,memVolumefraction_EA_20h_6_75','m
-o','LineWidth',1)
hold on

plot(memRadius_plot_normalized_0h_9_00,memVolumefraction_EA_0h_9_00','k-
s','LineWidth',1)
hold on
plot(memRadius_plot_normalized_1h_9_00,memVolumefraction_EA_1h_9_00','b-
s','LineWidth',1)
hold on
plot(memRadius_plot_normalized_5h_9_00,memVolumefraction_EA_5h_9_00','r-
s','LineWidth',1)
hold on
plot(memRadius_plot_normalized_20h_9_00,memVolumefraction_EA_20h_9_00','m
-s','LineWidth',1)
hold on
```

```
plot(memRadius_plot_normalized_0h_11_25,memVolumefraction_EA_0h_11_25','k
-^','LineWidth',1)
hold on
plot(memRadius_plot_normalized_1h_11_25,memVolumefraction_EA_1h_11_25','b
-^','LineWidth',1)
hold on
plot(memRadius_plot_normalized_5h_11_25,memVolumefraction_EA_5h_11_25','r
-^','LineWidth',1)
hold on
plot(memRadius_plot_normalized_20h_11_25,memVolumefraction_EA_20h_11_25',
'm-^','LineWidth',1)

% Label and Title of the plot
title('', 'FontSize',35)

% Normalized Radial Positions
xlabel('Normalized Radius', 'FontSize',20)

ylabel('\phi_{Ethyl acetate}', 'FontSize',20)
set(gca, 'FontSize', 20)
set(gca,'YTickLabel',{'0','0.10','0.20','0.30','0.40','0.50','0.60','0.70
',})

% Legend
legend('t = 0 h,   r_{3} = 6,75','t = 1 h,   r_{3} = 6,75','t = 5 h,
r_{3} = 6,75','t = 20 h, r_{3}  = 6,75','t = 0 h,   r_{3} = 9,00','t = 1
h,   r_{3} = 9,00','t = 5 h,   r_{3} = 9,00','t = 20 h, r_{3} = 9,00','t
= 0 h,   r_{3} = 11,25','t = 1 h,   r_{3} = 11,25','t = 5 h,   r_{3} =
11,25','t = 20 h, r_{3} = 11,25','Location','EastOutside')

% Define the intervall of the axis
axis([0 1 0.0 0.7])

% Resize the plot to fullscreen
set(gcf,'position',get(0,'screensize'))

% Save Figure as Volumefraction_EA.tiff
%saveas(gcf,'Volumefraction_EA.tiff', 'tiffn');
%--------------------------------------
```

## Volume_fraction_BA_Comparison.m

```
% Volume fraction BA Comparison
% (c) by Hannes Pucher
%-------------------------------------

%-------------------------------------
% Clear, Close and delete all
clc
clear all
close all
%-------------------------------------

%-------------------------------------
% Load files in the workspace

% r3=6.75
% Load memVolumefraction_BA_plot files
load('memVolumefraction_BA_plot_1h_6_75.mat')
memVolumefraction_BA_0h_6_75 = memVolumefraction_BA_plot(:,1);
load('memVolumefraction_BA_plot_1h_6_75.mat')
memVolumefraction_BA_1h_6_75 = memVolumefraction_BA_plot(:,11);
load('memVolumefraction_BA_plot_5h_6_75.mat')
memVolumefraction_BA_5h_6_75 = memVolumefraction_BA_plot(:,11);
load('memVolumefraction_BA_plot_20h_6_75.mat')
memVolumefraction_BA_20h_6_75 = memVolumefraction_BA_plot(:,11);

% Load memRadius_plot_normalized files
load('memRadius_plot_normalized_1h_6_75.mat')
memRadius_plot_normalized_0h_6_75 = memRadius_plot_normalized(:,1);
load('memRadius_plot_normalized_1h_6_75.mat')
memRadius_plot_normalized_1h_6_75 = memRadius_plot_normalized(:,11);
load('memRadius_plot_normalized_5h_6_75.mat')
memRadius_plot_normalized_5h_6_75 = memRadius_plot_normalized(:,11);
load('memRadius_plot_normalized_20h_6_75.mat')
memRadius_plot_normalized_20h_6_75 = memRadius_plot_normalized(:,11);
%-------------------------------------

%-------------------------------------
% r3=9.00
% Load memVolumefraction_BA_plot files
load('memVolumefraction_BA_plot_1h_9_00.mat')
memVolumefraction_BA_0h_9_00 = memVolumefraction_BA_plot(:,1);
load('memVolumefraction_BA_plot_1h_9_00.mat')
memVolumefraction_BA_1h_9_00 = memVolumefraction_BA_plot(:,11);
load('memVolumefraction_BA_plot_5h_9_00.mat')
memVolumefraction_BA_5h_9_00 = memVolumefraction_BA_plot(:,11);
load('memVolumefraction_BA_plot_20h_9_00.mat')
memVolumefraction_BA_20h_9_00 = memVolumefraction_BA_plot(:,11);

% Load memRadius_plot_normalized files
load('memRadius_plot_normalized_1h_9_00.mat')
memRadius_plot_normalized_0h_9_00 = memRadius_plot_normalized(:,1);
load('memRadius_plot_normalized_1h_9_00.mat')
memRadius_plot_normalized_1h_9_00 = memRadius_plot_normalized(:,11);
load('memRadius_plot_normalized_5h_9_00.mat')
memRadius_plot_normalized_5h_9_00 = memRadius_plot_normalized(:,11);
load('memRadius_plot_normalized_20h_9_00.mat')
```

```
memRadius_plot_normalized_20h_9_00 = memRadius_plot_normalized(:,11);
%--------------------------------------

%--------------------------------------
% r3=11.25
% Load memVolumefraction_BA_plot files
load('memVolumefraction_BA_plot_1h_11_25.mat')
memVolumefraction_BA_0h_11_25 = memVolumefraction_BA_plot(:,1);
load('memVolumefraction_BA_plot_1h_11_25.mat')
memVolumefraction_BA_1h_11_25 = memVolumefraction_BA_plot(:,11);
load('memVolumefraction_BA_plot_5h_11_25.mat')
memVolumefraction_BA_5h_11_25 = memVolumefraction_BA_plot(:,11);
load('memVolumefraction_BA_plot_20h_11_25.mat')
memVolumefraction_BA_20h_11_25 = memVolumefraction_BA_plot(:,11);

% Load memRadius_plot_normalized files
load('memRadius_plot_normalized_1h_11_25.mat')
memRadius_plot_normalized_0h_11_25 = memRadius_plot_normalized(:,1);
load('memRadius_plot_normalized_1h_11_25.mat')
memRadius_plot_normalized_1h_11_25 = memRadius_plot_normalized(:,11);
load('memRadius_plot_normalized_5h_11_25.mat')
memRadius_plot_normalized_5h_11_25 = memRadius_plot_normalized(:,11);
load('memRadius_plot_normalized_20h_11_25.mat')
memRadius_plot_normalized_20h_11_25 = memRadius_plot_normalized(:,11);
%--------------------------------------

%--------------------------------------
% Save all variables
save Experimental_Parameters.mat
%--------------------------------------

%--------------------------------------
% Plot Radius vs Volume fraction

figure(33)
plot(memRadius_plot_normalized_0h_6_75,memVolumefraction_BA_0h_6_75','k-
o','LineWidth',1)
hold on
plot(memRadius_plot_normalized_1h_6_75,memVolumefraction_BA_1h_6_75','b-
o','LineWidth',1)
hold on
plot(memRadius_plot_normalized_5h_6_75,memVolumefraction_BA_5h_6_75','r-
o','LineWidth',1)
hold on
plot(memRadius_plot_normalized_20h_6_75,memVolumefraction_BA_20h_6_75','m
-o','LineWidth',1)
hold on

plot(memRadius_plot_normalized_0h_9_00,memVolumefraction_BA_0h_9_00','k-
s','LineWidth',1)
hold on
plot(memRadius_plot_normalized_1h_9_00,memVolumefraction_BA_1h_9_00','b-
s','LineWidth',1)
hold on
plot(memRadius_plot_normalized_5h_9_00,memVolumefraction_BA_5h_9_00','r-
s','LineWidth',1)
hold on
plot(memRadius_plot_normalized_20h_9_00,memVolumefraction_BA_20h_9_00','m
-s','LineWidth',1)
hold on
```

```
plot(memRadius_plot_normalized_0h_11_25,memVolumefraction_BA_0h_11_25','k
-^','LineWidth',1)
hold on
plot(memRadius_plot_normalized_1h_11_25,memVolumefraction_BA_1h_11_25','b
-^','LineWidth',1)
hold on
plot(memRadius_plot_normalized_5h_11_25,memVolumefraction_BA_5h_11_25','r
-^','LineWidth',1)
hold on
plot(memRadius_plot_normalized_20h_11_25,memVolumefraction_BA_20h_11_25',
'm-^','LineWidth',1)

% Label and Title of the plot
title('', 'FontSize',35)

% Normalized Radial Positions
xlabel('Normalized Radius', 'FontSize',20)

ylabel('\phi_{Benzyl alcohol}', 'FontSize',20)
set(gca, 'FontSize', 20)
set(gca,'YTickLabel',{'0','0.05','0.10','0.15','0.20','0.25'})

% Legend
legend('t = 0 h,   r_{3} = 6,75','t = 1 h,   r_{3} = 6,75','t = 5 h,
r_{3} = 6,75','t = 20 h, r_{3}  = 6,75','t = 0 h,   r_{3} = 9,00','t = 1
h,   r_{3} = 9,00','t = 5 h,   r_{3} = 9,00','t = 20 h, r_{3} = 9,00','t
= 0 h,   r_{3} = 11,25','t = 1 h,   r_{3} = 11,25','t = 5 h,   r_{3} =
11,25','t = 20 h, r_{3} = 11,25','Location','EastOutside')

% Define the intervall of the axis
axis([0 1 0 0.25])

% Resize the plot to fullscreen
set(gcf,'position',get(0,'screensize'))

% Save Figure as Volumefraction_BA.tiff
%saveas(gcf,'Volumefraction_BA.tiff', 'tiffn');
%--------------------------------------
```

## Volume_fraction_RC_Comparison.m

```
% Volume fraction RC Comparison
% (c) by Hannes Pucher
%--------------------------------------

%--------------------------------------
% Clear, Close and delete all
clc
clear all
close all
%--------------------------------------

%--------------------------------------
% Load files in the workspace

% r3=6.75
% Load memVolumefraction_RC_plot files
load('memVolumefraction_RC_plot_1h_6_75.mat')
memVolumefraction_RC_0h_6_75 = volume_fraction_rest(:,1);
load('memVolumefraction_RC_plot_1h_6_75.mat')
memVolumefraction_RC_1h_6_75 = volume_fraction_rest(:,11);
load('memVolumefraction_RC_plot_5h_6_75.mat')
memVolumefraction_RC_5h_6_75 = volume_fraction_rest(:,11);
load('memVolumefraction_RC_plot_20h_6_75.mat')
memVolumefraction_RC_20h_6_75 = volume_fraction_rest(:,11);

% Load memRadius_plot_normalized files
load('memRadius_plot_normalized_1h_6_75.mat')
memRadius_plot_normalized_0h_6_75 = memRadius_plot_normalized(:,1);
load('memRadius_plot_normalized_1h_6_75.mat')
memRadius_plot_normalized_1h_6_75 = memRadius_plot_normalized(:,11);
load('memRadius_plot_normalized_5h_6_75.mat')
memRadius_plot_normalized_5h_6_75 = memRadius_plot_normalized(:,11);
load('memRadius_plot_normalized_20h_6_75.mat')
memRadius_plot_normalized_20h_6_75 = memRadius_plot_normalized(:,11);
%--------------------------------------

%--------------------------------------
% r3=9.00
% Load volume_fraction_RC files
load('memVolumefraction_RC_plot_1h_9_00.mat')
memVolumefraction_RC_0h_9_00 = volume_fraction_rest(:,1);
load('memVolumefraction_RC_plot_1h_9_00.mat')
memVolumefraction_RC_1h_9_00 = volume_fraction_rest(:,11);
load('memVolumefraction_RC_plot_5h_9_00.mat')
memVolumefraction_RC_5h_9_00 = volume_fraction_rest(:,11);
load('memVolumefraction_RC_plot_20h_9_00.mat')
memVolumefraction_RC_20h_9_00 = volume_fraction_rest(:,11);

% Load memRadius_plot_normalized files
load('memRadius_plot_normalized_1h_9_00.mat')
memRadius_plot_normalized_0h_9_00 = memRadius_plot_normalized(:,1);
load('memRadius_plot_normalized_1h_9_00.mat')
memRadius_plot_normalized_1h_9_00 = memRadius_plot_normalized(:,11);
load('memRadius_plot_normalized_5h_9_00.mat')
memRadius_plot_normalized_5h_9_00 = memRadius_plot_normalized(:,11);
load('memRadius_plot_normalized_20h_9_00.mat')
```

```
memRadius_plot_normalized_20h_9_00 = memRadius_plot_normalized(:,11);
%--------------------------------------

%--------------------------------------
% r3=11.25
% Load volume_fraction_RC files
load('memVolumefraction_RC_plot_1h_11_25.mat')
memVolumefraction_RC_0h_11_25 = volume_fraction_Rest(:,1);
load('memVolumefraction_RC_plot_1h_11_25.mat')
memVolumefraction_RC_1h_11_25 = volume_fraction_Rest(:,11);
load('memVolumefraction_RC_plot_5h_11_25.mat')
memVolumefraction_RC_5h_11_25 = volume_fraction_rest(:,11);
load('memVolumefraction_RC_plot_20h_11_25.mat')
memVolumefraction_RC_20h_11_25 = volume_fraction_rest(:,11);

% Load memRadius_plot_normalized files
load('memRadius_plot_normalized_1h_11_25.mat')
memRadius_plot_normalized_0h_11_25 = memRadius_plot_normalized(:,1);
load('memRadius_plot_normalized_1h_11_25.mat')
memRadius_plot_normalized_1h_11_25 = memRadius_plot_normalized(:,11);
load('memRadius_plot_normalized_5h_11_25.mat')
memRadius_plot_normalized_5h_11_25 = memRadius_plot_normalized(:,11);
load('memRadius_plot_normalized_20h_11_25.mat')
memRadius_plot_normalized_20h_11_25 = memRadius_plot_normalized(:,11);
%--------------------------------------

%--------------------------------------
% Save all variables
%save Experimental_Parameters.mat
%--------------------------------------

%--------------------------------------
% Plot Radius vs Volume fraction

figure(33)
plot(memRadius_plot_normalized_0h_6_75,memVolumefraction_RC_0h_6_75','k-
o','LineWidth',1)
hold on
plot(memRadius_plot_normalized_1h_6_75,memVolumefraction_RC_1h_6_75','b-
o','LineWidth',1)
hold on
plot(memRadius_plot_normalized_5h_6_75,memVolumefraction_RC_5h_6_75','r-
o','LineWidth',1)
hold on
plot(memRadius_plot_normalized_20h_6_75,memVolumefraction_RC_20h_6_75','m
-o','LineWidth',1)
hold on

plot(memRadius_plot_normalized_0h_9_00,memVolumefraction_RC_0h_9_00','k-
s','LineWidth',1)
hold on
plot(memRadius_plot_normalized_1h_9_00,memVolumefraction_RC_1h_9_00','b-
s','LineWidth',1)
hold on
plot(memRadius_plot_normalized_5h_9_00,memVolumefraction_RC_5h_9_00','r-
s','LineWidth',1)
hold on
plot(memRadius_plot_normalized_20h_9_00,memVolumefraction_RC_20h_9_00','m
-s','LineWidth',1)
hold on
```

```
plot(memRadius_plot_normalized_0h_11_25,memVolumefraction_RC_0h_11_25','k
-^','LineWidth',1)
hold on
plot(memRadius_plot_normalized_1h_11_25,memVolumefraction_RC_1h_11_25','b
-^','LineWidth',1)
hold on
plot(memRadius_plot_normalized_5h_11_25,memVolumefraction_RC_5h_11_25','r
-^','LineWidth',1)
hold on
plot(memRadius_plot_normalized_20h_11_25,memVolumefraction_RC_20h_11_25',
'm-^','LineWidth',1)

% Label and Title of the plot
title('', 'FontSize',35)

% Normalized Radial Positions
xlabel('Normalized Radius', 'FontSize',20)

ylabel('\phi_{Remaining Components}', 'FontSize',20)
set(gca, 'FontSize', 20)
set(gca,'YTickLabel',{'0','0.10','0.20','0.30','0.40','0.50','0.60','0.70
','0.80','0.90','1.00'})

% Legend
legend('t = 0 h,   r_{3} = 6,75','t = 1 h,   r_{3} = 6,75','t = 5 h,
r_{3} = 6,75','t = 20 h, r_{3}  = 6,75','t = 0 h,   r_{3} = 9,00','t = 1
h,   r_{3} = 9,00','t = 5 h,   r_{3} = 9,00','t = 20 h, r_{3} = 9,00','t
= 0 h,   r_{3} = 11,25','t = 1 h,   r_{3} = 11,25','t = 5 h,   r_{3} =
11,25','t = 20 h, r_{3} = 11,25','Location','EastOutside')

% Define the intervall of the axis
axis([0 1 0 1])

% Resize the plot to fullscreen
%set(gcf,'position',get(0,'screensize'))

% Save Figure as Volumefraction_RC.tiff
% saveas(gcf,'Volumefraction_RC.tiff', 'tiffn');
%--------------------------------------

%--------------------------------------
% Diffusion Coefficient

volume_f_initial_API = 0.0412;

% System depending Components

r_BA_1 = 1.6916;
r_EA_1 = 1.7496;


% r3 = 6.75
r_BA_2_6 = 6.2479;
r_BA_3_6 = 6.7521;
Diffusion_BA_6_0h = r_BA_1*10.^(-r_BA_2_6 -
(memVolumefraction_RC_0h_11_25-volume_f_initial_API).*r_BA_3_6);
Diffusion_BA_6_1h = r_BA_1*10.^(-r_BA_2_6 -
(memVolumefraction_RC_1h_11_25-volume_f_initial_API).*r_BA_3_6);
```

```
Diffusion_BA_6_5h = r_BA_1*10.^(-r_BA_2_6 -
(memVolumefraction_RC_5h_11_25-volume_f_initial_API).*r_BA_3_6);
Diffusion_BA_6_20h = r_BA_1*10.^(-r_BA_2_6 -
(memVolumefraction_RC_20h_11_25-volume_f_initial_API).*r_BA_3_6);
r_EA_2_6 = 6.2479;
r_EA_3_6 = 6.7521;
Diffusion_EA_6_0h = r_EA_1*10.^(-r_EA_2_6 -
(memVolumefraction_RC_0h_11_25-volume_f_initial_API).*r_EA_3_6);
Diffusion_EA_6_1h = r_EA_1*10.^(-r_EA_2_6 -
(memVolumefraction_RC_1h_11_25-volume_f_initial_API).*r_EA_3_6);
Diffusion_EA_6_5h = r_EA_1*10.^(-r_EA_2_6 -
(memVolumefraction_RC_5h_11_25-volume_f_initial_API).*r_EA_3_6);
Diffusion_EA_6_20h = r_EA_1*10.^(-r_EA_2_6 -
(memVolumefraction_RC_20h_11_25-volume_f_initial_API).*r_EA_3_6);

% r3 = 9.00
r_BA_2_9 = 5.9972;
r_BA_3_9 = 9.0028;
Diffusion_BA_9_0h = r_BA_1*10.^(-r_BA_2_9 -
(memVolumefraction_RC_0h_11_25-volume_f_initial_API).*r_BA_3_9);
Diffusion_BA_9_1h = r_BA_1*10.^(-r_BA_2_9 -
(memVolumefraction_RC_1h_11_25-volume_f_initial_API).*r_BA_3_9);
Diffusion_BA_9_5h = r_BA_1*10.^(-r_BA_2_9 -
(memVolumefraction_RC_5h_11_25-volume_f_initial_API).*r_BA_3_9);
Diffusion_BA_9_20h = r_BA_1*10.^(-r_BA_2_9 -
(memVolumefraction_RC_20h_11_25-volume_f_initial_API).*r_BA_3_9);
r_EA_2_9 = 5.9972;
r_EA_3_9 = 9.0028;
Diffusion_EA_9_0h = r_EA_1*10.^(-r_EA_2_9 -
(memVolumefraction_RC_0h_11_25-volume_f_initial_API).*r_EA_3_9);
Diffusion_EA_9_1h = r_EA_1*10.^(-r_EA_2_9 -
(memVolumefraction_RC_1h_11_25-volume_f_initial_API).*r_EA_3_9);
Diffusion_EA_9_5h = r_EA_1*10.^(-r_EA_2_9 -
(memVolumefraction_RC_5h_11_25-volume_f_initial_API).*r_EA_3_9);
Diffusion_EA_9_20h = r_EA_1*10.^(-r_EA_2_9 -
(memVolumefraction_RC_20h_11_25-volume_f_initial_API).*r_EA_3_9);

% r3 = 11.25
r_BA_2_11 = 5.7465;
r_BA_3_11 = 11.2535;
Diffusion_BA_11_0h = r_BA_1*10.^(-r_BA_2_11 -
(memVolumefraction_RC_0h_11_25-volume_f_initial_API).*r_BA_3_11);
Diffusion_BA_11_1h = r_BA_1*10.^(-r_BA_2_11 -
(memVolumefraction_RC_1h_11_25-volume_f_initial_API).*r_BA_3_11);
Diffusion_BA_11_5h = r_BA_1*10.^(-r_BA_2_11 -
(memVolumefraction_RC_5h_11_25-volume_f_initial_API).*r_BA_3_11);
Diffusion_BA_11_20h = r_BA_1*10.^(-r_BA_2_11 -
(memVolumefraction_RC_20h_11_25-volume_f_initial_API).*r_BA_3_11);

r_EA_2_11 = 5.7465;
r_EA_3_11 = 11.2535;
Diffusion_EA_11_0h = r_EA_1*10.^(-r_EA_2_11 -
(memVolumefraction_RC_0h_11_25-volume_f_initial_API).*r_EA_3_11);
Diffusion_EA_11_1h = r_EA_1*10.^(-r_EA_2_11 -
(memVolumefraction_RC_1h_11_25-volume_f_initial_API).*r_EA_3_11);
Diffusion_EA_11_5h = r_EA_1*10.^(-r_EA_2_11 -
(memVolumefraction_RC_5h_11_25-volume_f_initial_API).*r_EA_3_11);
Diffusion_EA_11_20h = r_EA_1*10.^(-r_EA_2_11 -
(memVolumefraction_RC_20h_11_25-volume_f_initial_API).*r_EA_3_11);
%--------------------------------------
```

```
%--------------------------------------
% Plot Radius vs Diffusion Coefficient for EA and BA r3=6.75
figure(40)

semilogy(memRadius_plot_normalized_0h_6_75,Diffusion_EA_6_0h','k-
^','LineWidth',1)
hold on
semilogy(memRadius_plot_normalized_1h_6_75,Diffusion_EA_6_1h','b-
^','LineWidth',1)
hold on
semilogy(memRadius_plot_normalized_5h_6_75,Diffusion_EA_6_5h','r-
^','LineWidth',1)
hold on
semilogy(memRadius_plot_normalized_20h_6_75,Diffusion_EA_6_20h','m-
^','LineWidth',1)
hold on

semilogy(memRadius_plot_normalized_0h_6_75,Diffusion_BA_6_0h','k-
o','LineWidth',1)
hold on
semilogy(memRadius_plot_normalized_1h_6_75,Diffusion_BA_6_1h','b-
o','LineWidth',1)
hold on
semilogy(memRadius_plot_normalized_5h_6_75,Diffusion_BA_6_5h','r-
o','LineWidth',1)
hold on
semilogy(memRadius_plot_normalized_20h_6_75,Diffusion_BA_6_20h','m-
o','LineWidth',1)
hold on

% Label and Title of the plot
title('', 'FontSize',35)

% Normalized Radial Positions
xlabel('Normalized Radius', 'FontSize',20)

ylabel('Diffusion Coefficient', 'FontSize',20)
set(gca, 'FontSize', 20)
%set(gca,'YTickLabel',{'0','0.10','0.20','0.30','0.40','0.50','0.60','0.7
0','0.80','0.90','1.00'})

% Legend
legend('D_{EA,t=0h}','D_{EA,t=1h}','D_{EA,t=5h}','D_{EA,t=20h}','D_{BA,t=
0h}','D_{BA,t=1h}','D_{BA,t=5h}','D_{BA,t=20h}','Location','East')

% Define the intervall of the axis
%axis([0 1 0 1])

% Resize the plot to fullscreen
set(gcf,'position',get(0,'screensize'))

% Save Figure as Diffusion_6.75.tiff
% saveas(gcf,'Diffusion_6.75.tiff', 'tiffn');
%--------------------------------------


%--------------------------------------
% Plot Radius vs Diffusion Coefficient for EA and BA r3=9.00
```

```
figure(41)

semilogy(memRadius_plot_normalized_0h_9_00,Diffusion_EA_9_0h','k-
^','LineWidth',1)
hold on
semilogy(memRadius_plot_normalized_1h_9_00,Diffusion_EA_9_1h','b-
^','LineWidth',1)
hold on
semilogy(memRadius_plot_normalized_5h_9_00,Diffusion_EA_9_5h','r-
^','LineWidth',1)
hold on
semilogy(memRadius_plot_normalized_20h_9_00,Diffusion_EA_9_20h','m-
^','LineWidth',1)
hold on

semilogy(memRadius_plot_normalized_0h_9_00,Diffusion_BA_9_0h','k-
^','LineWidth',1)
hold on
semilogy(memRadius_plot_normalized_1h_9_00,Diffusion_BA_9_1h','b-
o','LineWidth',1)
hold on
semilogy(memRadius_plot_normalized_5h_9_00,Diffusion_BA_9_5h','r-
o','LineWidth',1)
hold on
semilogy(memRadius_plot_normalized_20h_9_00,Diffusion_BA_9_20h','m-
o','LineWidth',1)
hold on

% Label and Title of the plot
title('', 'FontSize',35)

% Normalized Radial Positions
xlabel('Normalized Radius', 'FontSize',20)

ylabel('Diffusion Coefficient', 'FontSize',20)
set(gca, 'FontSize', 20)
%set(gca,'YTickLabel',{'0','0.10','0.20','0.30','0.40','0.50','0.60','0.7
0','0.80','0.90','1.00'})

% Legend
legend('D_{EA,t=0h}','D_{EA,t=1h}','D_{EA,t=5h}','D_{EA,t=20h}','D_{BA,t=
0h}','D_{BA,t=1h}','D_{BA,t=5h}','D_{BA,t=20h}','Location','East')

% Define the intervall of the axis
%axis([0 1 0 1])

% Resize the plot to fullscreen
set(gcf,'position',get(0,'screensize'))

% Save Figure as Diffusion_9.00.tiff
% saveas(gcf,'Diffusion_9.00.tiff', 'tiffn');
%--------------------------------------
%--------------------------------------

%--------------------------------------
% Plot Radius vs Diffusion Coefficient for EA and BA r3=11.25
figure(42)
```

```
semilogy(memRadius_plot_normalized_0h_11_25,Diffusion_EA_11_0h','k-
^','LineWidth',1)
hold on
semilogy(memRadius_plot_normalized_1h_11_25,Diffusion_EA_11_1h','b-
^','LineWidth',1)
hold on
semilogy(memRadius_plot_normalized_5h_11_25,Diffusion_EA_11_5h','r-
^','LineWidth',1)
hold on
semilogy(memRadius_plot_normalized_20h_11_25,Diffusion_EA_11_20h','m-
^','LineWidth',1)
hold on

semilogy(memRadius_plot_normalized_0h_11_25,Diffusion_BA_11_0h','k-
o','LineWidth',1)
hold on
semilogy(memRadius_plot_normalized_1h_11_25,Diffusion_BA_11_1h','b-
o','LineWidth',1)
hold on
semilogy(memRadius_plot_normalized_5h_11_25,Diffusion_BA_11_5h','r-
o','LineWidth',1)
hold on
semilogy(memRadius_plot_normalized_20h_11_25,Diffusion_BA_11_20h','m-
o','LineWidth',1)
hold on

% Label and Title of the plot
title('', 'FontSize',35)

% Normalized Radial Positions
xlabel('Normalized Radius', 'FontSize',20)

ylabel('Diffusion Coefficient', 'FontSize',20)
set(gca, 'FontSize', 20)
%set(gca,'YTickLabel',{'0','0.10','0.20','0.30','0.40','0.50','0.60','0.7
0','0.80','0.90','1.00'})

% Legend
legend('D_{EA,t=0h}','D_{EA,t=1h}','D_{EA,t=5h}','D_{EA,t=20h}','D_{BA,t=
0h}','D_{BA,t=1h}','D_{BA,t=5h}','D_{BA,t=20h}','Location','East')

% Define the intervall of the axis
%axis([0 1 0 1])

% Resize the plot to fullscreen
set(gcf,'position',get(0,'screensize'))

% Save Figure as Diffusion_11.25.tiff
% saveas(gcf,'Diffusion_11.25.tiff', 'tiffn');
%--------------------------------------
```

## D_Variations.m

```
% D Variations
% (c) by Hannes Pucher
%-------------------------------------

%-------------------------------------
% Clear, Close and delete all
clc
clear all
close all
%-------------------------------------

%-------------------------------------
% Load files in the workspace

% Load memRadius_plot files
load('memRadius_end_d50.mat')
memRadius_d50 = memRadius_end*10^6;
load('memRadius_end_d100.mat')
memRadius_d100 = memRadius_end*10^6;
load('memRadius_end_d163.mat')
memRadius_d163 = memRadius_end*10^6;
load('memRadius_end_d200.mat')
memRadius_d200 = memRadius_end*10^6;
%-------------------------------------

%-------------------------------------
% Calculate the Percentage of Shrinkage
Percent_d50 = 100-(memRadius_d50*2)/(memRadius_d50(1,1)*2)*100;
Percent_d100 = 100-(memRadius_d100*2)/(memRadius_d100(1,1)*2)*100;
Percent_d163 = 100-(memRadius_d163*2)/(memRadius_d163(1,1)*2)*100;
Percent_d200 = 100-(memRadius_d200*2)/(memRadius_d200(1,1)*2)*100;
%-------------------------------------

%-------------------------------------
% Define the memTime_plot vector for the plot
memTime_plot = linspace(0,3601,3601);
%-------------------------------------

%-------------------------------------
% Save all variables
save Experimental_Parameters.mat
%-------------------------------------

%-------------------------------------
% Plot Radius vs Time
figure(1)
plot(memTime_plot/3600,memRadius_d50*2,'k','LineWidth',2)
hold on
plot(memTime_plot/3600,memRadius_d100*2,'m-.','LineWidth',2)
hold on
plot(memTime_plot/3600,memRadius_d163*2,'r--','LineWidth',2)
hold on
plot(memTime_plot/3600,memRadius_d200*2,'b:','LineWidth',2)

% Label and Title of the plot
title('', 'FontSize',35)
```

```
xlabel('Time [h]', 'FontSize',20)
ylabel('SMD [\mum]', 'FontSize',20)
set(gca, 'FontSize', 20)

% Legend
legend('d_{t=0} = 50','d_{t=0} = 100','d_{t=0} = 163','d_{t=0} =
200','Location','NorthEast')

% Define the intervall of the axis
axis([0 1 0 250])

% Resize the plot to fullscreen
set(gcf,'position',get(0,'screensize'))

% Save Figure as SMD.tiff
saveas(gcf,'SMD.tiff', 'tiffn');
%--------------------------------------

%--------------------------------------
% Plot Percentage of Shrinkage vs Time
figure(11)
plot(memTime_plot/3600,Percent_d50,'k','LineWidth',2)
hold on
plot(memTime_plot/3600,Percent_d100,'m-.','LineWidth',2)
hold on
plot(memTime_plot/3600,Percent_d163,'r--','LineWidth',2)
hold on
plot(memTime_plot/3600,Percent_d200,'b:','LineWidth',2)

% Label and Title of the plot
title('', 'FontSize',35)
xlabel('Time [h]', 'FontSize',20)
ylabel('Shrinkage [%]', 'FontSize',20)
set(gca, 'FontSize', 20)

% Legend
legend('d_{t=0} = 50','d_{t=0} = 100','d_{t=0} = 163','d_{t=0} =
200','Location','SouthEast')

% Define the intervall of the axis
axis([0 1 0 50])

% Resize the plot to fullscreen
set(gcf,'position',get(0,'screensize'))

% Save Figure as SMD.tiff
saveas(gcf,'Percentage.tiff', 'tiffn');
%--------------------------------------
```